ARI Contractor Report 96-58

Design Plan for Collection, Storage, and Access of Close Air Support Performance Data: Air-Ground Training and Feedback System (AGTFS)

James A. Huffman, David Butterfield, and Paul A. Jarrett

BDM Federal, Inc.

August 1996

United States Army Research Institute for the Behavioral and Social Sciences

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE August 1996	3. REPORT TYPE AND DATES COVERED Interim Report 07/17/92 - 10/16/94

4. TITLE AND SUBTITLE Design Plan for Collection, Storage, and Access of Close Air Support Performance Data: Air-Ground Training and Feedback System (AGTFS)	5. FUNDING NUMBERS MDA903-92-D-0075-0006 3414 CO3 665803 D730
6. AUTHOR(S) James A. Huffman David Butterfield Paul A. Jarrett	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) BDM FEDERAL INC. DOD CENTER MONTEREY BAY 400 GIGLING ROAD SEASIDE, CA 93955	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. ARMY RESEARCH INSTITUTE FOR THE BEHAVIORAL AND SOCIAL SCIENCES 5001 EISENHOWER AVENUE ALEXANDRIA, VA 22333-5600	10. SPONSORING/MONITORING AGENCY REPORT NUMBER Contractor Report 96-58

11. SUPPLEMENTARY NOTES
The COR is Michael R. McCluskey. This report is published to meet legal and contractual requirements and may not meet ARI's scientific or professional standards for publication.

12a. DISTRIBUTION/AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.	12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

As the Airland Battle doctrine continues to evolve, the need for enhanced coordination between the ground and air forces becomes ever more critical. To meet this requirement the military has instituted a number of organizational reforms, such as unified commands and an emphasis on staff training in interservice operations. The Gulf War further highlighted the need for increased training at the tactical level.

The increased emphasis on joint operations and the availability of the Combat Training Centers that employ Airland Battle scenarios make it both practical and worthwhile to develop an air-ground training and feedback system. The AGTFS provides for the results to be organized into a common database. This will allow all services access to information to identify systemic issues, guide training development, and provide feedback to units on their performance during training. The data collection will provide a quantifiable basis for further training resource requirements.

14. SUBJECT TERMS Electronic Collection Instrument (ECI), Lethality Component Measure (LCM), Observer/Controllers (O/Cs), Survivability Component Measure (SCM), Contribution Component Measure (CCM)	15. NUMBER OF PAGES 16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UNLIMITED

CONTRACT #: MDA 903-92-D-0075
DELIVERABLE #: 0006-04-011
SEPTEMBER 1, 1994

Design Plan for Collection, Storage, and Access of Close Air Support Performance Data

Air-Ground Training and Feedback System (AGTFS)

James A. Huffman
PRC, Inc.

David Butterfield
PRC, Inc.

Paul A. Jarrett
HumRRO

Submitted by: Mr. Michael R. McCluskey, Acting Chief
Unit-Collective Training Research Unit
and Jack Hiller, Director
Training Research Laboratory

Mr. Michael R. McCluskey, Contracting Officer's Representative

January 5, 1995

U.S. Army Research Institute

DESIGN PLAN FOR THE COLLECTION, STORAGE, AND ACCESS OF CLOSE AIR SUPPORT PERFORMANCE DATA

Page

LIST OF FIGURES

Page

LIST OF FIGURES (Cont.)

LIST OF TABLES

LIST OF APPENDICES

DESIGN PLAN FOR THE COLLECTION, STORAGE, AND ACCESS OF CLOSE AIR SUPPORT PERFORMANCE DATA FOR THE AIR-GROUND TRAINING AND FEEDBACK SYSTEM (AGTFS)

I. PURPOSE

This document is the eighth in a series of interim reports concerning the Air-Ground Training Feedback System (AGTFS) being developed under Army Research Institute Contract MDA 903-92-D-0075. The purpose of this paper is to describe the design and specifications of the Close Air Support database as well as how the measurement data, both process and outcome, for close air support (CAS) in a mid-high intensity conflict will be collected and entered into the research database.

II. INTRODUCTION

As the Airland Battle doctrine continues to evolve, the need for enhanced coordination between the ground and air forces becomes ever more critical. To meet this requirement the military has instituted a number of organizational reforms, such as unified commands and an emphasis on staff training in interservice operations. The Gulf War further highlighted the need for increased training at the tactical level.

Joint ground-air training is conducted at various levels at all the Combat Training Centers (CTCs). The National Training Center (NTC) is linked to the Air Forces's Air Warrior I program which supplies CAS in a mid-high intensity combat training environment. The JRTC is supported by Air Warrior II which provides CAS in a Low Intensity Conflict (LIC) environment. The training environments at both locations provide adequate maneuver and air space to provide realistic training to both ground and air forces.

The increased emphasis on joint operations and the availability of the Combat Training Centers that employ Airland Battle scenarios make it both practical and worthwhile to develop an air-ground training and feedback system. The AGTFS provides for the results to be organized into a common database. This will allow all services access to information to identify systemic issues, guide training development, and provide feedback to units on their performance during training. The data collected will provide a quantifiable basis for further training resource requirements.

Several previously published reports in this study, and in the parallel CAS in low intensity conflict study, have provided necessary input to the database. These include: <u>Measurement Model for Evaluating Mission Results and Evaluating CAS Battle Task Performance</u> (Keesling 1993), <u>Integrated List of Battle Tasks for Close Air Support in a Mid to High Intensity Conflict</u> (Root, 1994), and <u>Performance Measurement Index</u> (Jarrett, 1994), <u>Field Tryout of Close Air</u>

Support Task Performance Measurement System, Air-Ground Training and Feedback System (Huffman 1994), and Database Specification Report, Air Ground training and Feedback System for Low Intensity Conflict (Butterfield, Huffman, Jarrett, 1994). The paragraphs below contain a more detailed discussion of the applicability of these five reports to the database and this report.

The first report describes the measurement model used for this project which is discussed in the Measurement System Criteria Model paragraph below.

The second report identified the battle tasks necessary to accomplish CAS in a mid-high intensity environment. These provided the basis for the process measures that were used to collect unit performance data during the field tryout.

The third report (Performance Measurement Index) identified the instrumented systems in place at NTC and Air Warrior I. It developed the outcome measures based on data that can be collected using the existing instrumentation and addressed the methodology, procedures, and equipment for collecting outcome measures at NTC. This collection system and the outcome measures are discussed in detail in the Data Collection section, below.

The fourth report (Field Tryout of Close Air Support Task Performance Measurement System) addressed the field verification test conducted at NTC. This test evaluated the process and outcome measures and the applicability of the tasks in the task list. The task list for the NTC evaluation was modified based on the results of the, previously conducted, JRTC field verification. After modification, the list was reconfigured into forms suitable for the assessment of CAS employment during a field exercise at the NTC. The reconfigured task list was used to assess the employment of CAS, and the applicability of the tasks themselves, during the required field test of the measurement system. Using the results of the field verification test, the CAS task list was again modified and a final list was produced.

The fifth report (Database Specification Report) describes the development of the prototype database. That database was used to collect and analyze the data from the verification test conducted at JRTC and the field tryout test at NTC. The database was constructed based on the identified data collection mechanisms and sample report requirements, as described in the Database Specification Report.

III. STUDY METHODOLOGY

Using the model constructed for the Air Ground Training Feedback System, Mid-High Intensity study as a framework, the CAS battle tasks were identified. A front end analysis was conducted to identify the specific criteria measures of battle task performance. A doctrinal review was conducted and the tasks derived were used to produce a doctrinally based task list.

The candidate task list was then verified through interviews with subject matter experts (SMEs) at NTC and AW1. The tasks were modified based on those interviews and reconfigured into forms suitable for the assessment of CAS employment during an exercise. The reconfigured task list was used to assess the employment of CAS, and the applicability of the tasks themselves, during the required field tryout of the measurement system. Using the results of the field tryout, the CAS task list was modified and a final list was produced.

While the field tryout at NTC was being conducted, the prototype database constructed during the field verification of the low intensity CAS task list was modified to provide for the different data collection mechanisms at NTC and to allow input of automated outcome measure data. Once the field tryout at NTC was completed, a final, tested, CAS battle task list was developed that consolidated the battle tasks for both the mid-high intensity and the low intensity conflict. A new database structure was then constructed which would apply to that finalized, consolidated, task list.

The specifications and the programs identified in this report are those that describe the database that resulted from the use of the prototype database to compile and analyze the data derived from the field tryout test at NTC. The major changes to the prototype database were a reordering of the performance tasks (process measures), elimination of several tasks, deletion of some variables and/or report selection criteria from the database, deletion of one pre-set report, finalization of the outcome measures report formats, and the addition of Maneuver Execution tasks. The development and specifications of that database are described below.

IV. MEASUREMENT SYSTEM CRITERIA MODEL

A Measurement Model Supporting the Air-Ground Training Feedback System (Keesling, 1993) reported on the model developed for the Air-Ground Training Feedback System for Close Air Support study (Root, 1994). The measurement model identified two main areas in which performance must be measured. The first was "process measures" which examine the tasks that commanders and units must perform to synchronize and apply the air-ground assets effectively. The second was "outcome measures" which examine the effects of the air-ground systems on the battlefield.

Figure 1 shows a conceptualization of the measurement model. The figure depicts the use of doctrinal and operational information sources to develop process measures. These process measures are in the form of battle tasks which must be performed by both the ground and air forces involved in planning and execution of CAS. The performance measures are linked to the outcome measures with an arrow showing that variations in the performance of process tasks will result in variations in outcomes. The results of both the process and the outcome measures are then shown feeding into four separate areas or functions: providing AAR input to ground/air units; providing lessons learned for the ground/air units; making revisions to the measurement system; and, building a research database to provide data for doctrine, organization, training, materiel, and leadership (DOTML) decisions.

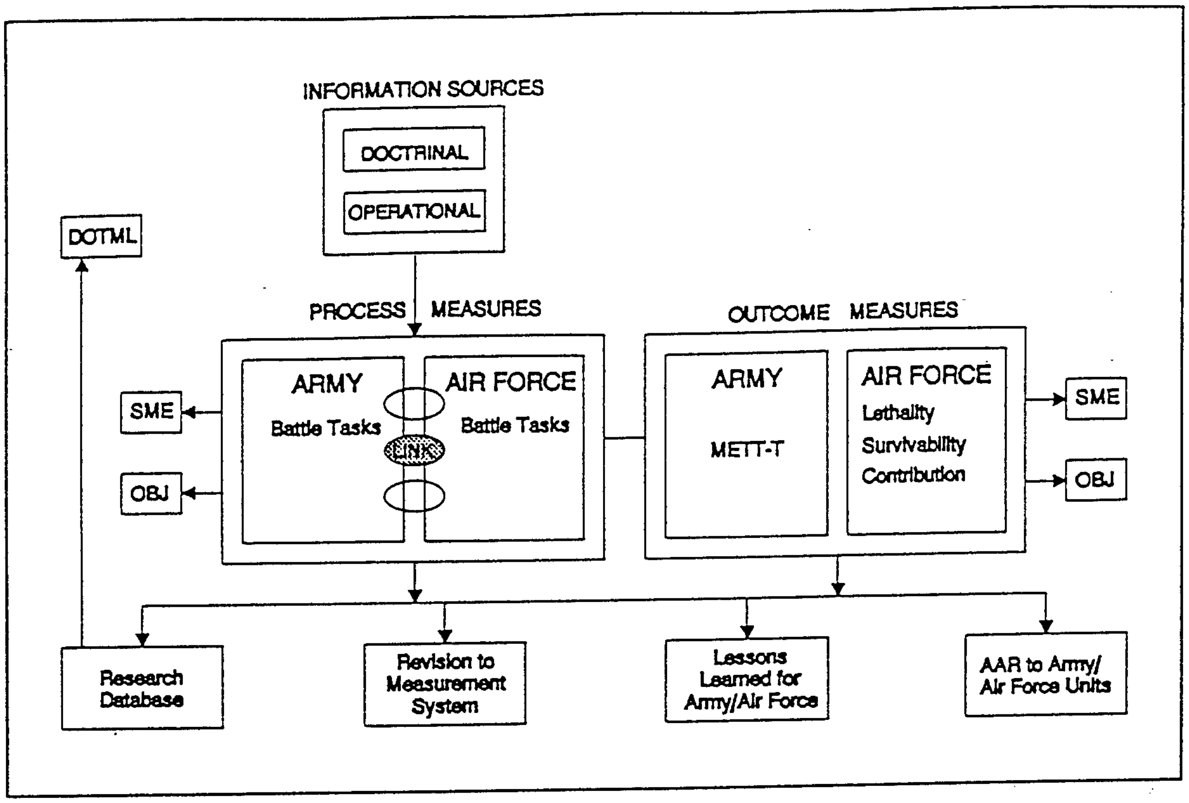

Figure 1 Schematic Organization of the Battle Task and Outcome Measurement Model

This model is applicable to both mid-high intensity and low intensity conflict and provides an operational structure which organizes the CAS measurement system. The model identified broad categories of measures to be included. Previous research (List of Battle Tasks for Close Air Support in a Low Intensity Conflict (Front End Analysis) (Huffman 1994) and Field Tryout of Close Air Support Battle Task Measurement System (Implementation Test Report) (Huffman 1994)) demonstrated the applicability of methods for developing specific process and outcome measures. An additional consideration was the current training feedback processes (After Action Reviews, Mission Debriefs, etc.) being used at the NTC, Fort Irwin, CA, and Air Warrior I, Nellis AFB, Nevada. Process measures include performance of the close air support battle tasks (in planning, preparation, and execution phases). Outcome measures include tactical mission success or failure, bomb damage, aircraft losses etc. Data on mission conditions (weather, terrain, type mission, etc.) are also needed to provide a context for the other measures. Analysis of these measures indicated that some could be highly objective (e.g., losses of combat systems) while others would require judgement by subject matter experts (e.g., performance of certain battle tasks).

Using this analysis framework, the data to be collected and stored in the database was identified. The data format of the task list used to collect data on CAS in the mid-high intensity environment at NTC was designed to be compatible with the Combined Arms Battle Task mission books (Lewman, 1994) as well as other analytical tools in use at the Combat Training Center Archive.

IV. DATABASE REQUIREMENTS

A. General Requirements: Defined in Statement of Work

The development of the database for the Air Ground Training and Feedback System (AGTFS) is closely coordinated with the study of close air support for low intensity conflict (Root, 1994). The separate statements of work for both projects required each resulting database to allow "researchers to fully utilize the data collected with this methodology," and to "provide a structure that will accommodate the storage and retrieval of this data." They further state, "This data should be readily available to the user community and accessible at the CTC Archive." This requirement was interpreted to mean that data should be gathered and stored in a common system that would conform to database standards developed for the CTC Archive and would facilitate linking the data in the CAS databases with other data in the archive.

By using a common CAS measurement framework (Keesling, 1993), developing very similar data collection instruments, and by designing a common database, the two AGTFS projects were able to conform to the requirements for user access and compatibility with CTC Archive databases while minimizing redundant development efforts. The common database, called the Collective AGTFS Database, is designed to be used in conjunction with other CTC Archive databases as shown in Figure 2. At each CTC shown, the information gathered about CAS-related performance may be fed into the AAR and THP products developed at the CTC (e.g., task performance) while other information that is collected for those products is fed into the collection of CAS performance measures (e.g., air BDA). The AAR and THP products are intended to inform the rotating units about their performance so they may improve upon their training.

Once the CAS performance measures from each CTC are entered into the Collective AGTFS Database, they may then be combined with information from other CTC Archive databases to develop lessons learned and various research products (e.g., trends studies or studies of particular DOTML issues). Not shown in Figure 2, for the sake of clarity, are the linkages fron AAR and THP products to the other CTC Archive databases, or the additional data collection effort that go into filling those databases.

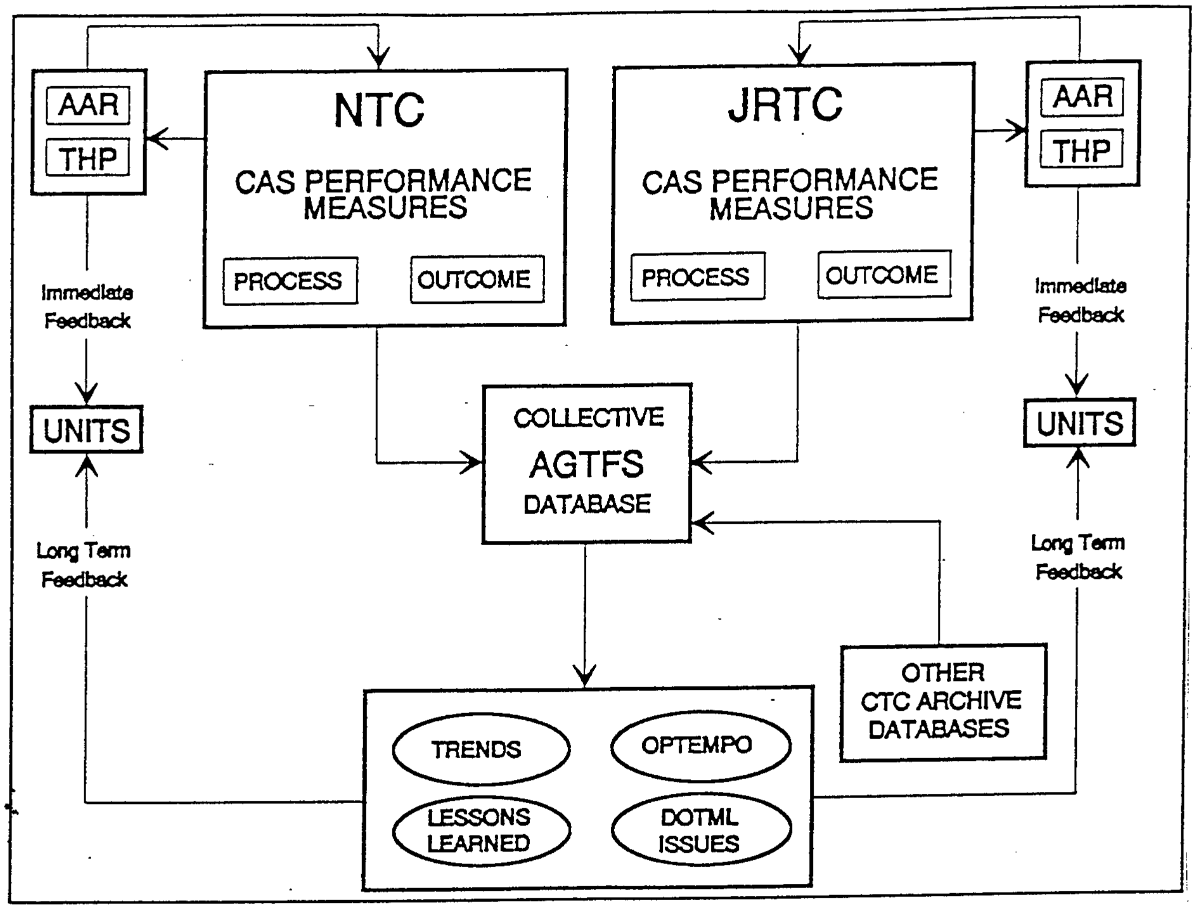

Figure 2 Schematic of Collective Database Position within the AGTFS

VI. DATABASE REPORT

A. Formats and Organizational Requirements

The initial database requirements were identified prior to the field verification test of the close air support battle task list at JRTC. Specific pre-formatted database reports that were anticipated to be required for the analysis of the results of the field verification test were identified. As a result of identifying those report formats, the need to be able to sort the database by several different variables was also identified.

As the development progressed, it was evident that several of the identified formats were too cumbersome to be of much use. In addition, as the field verification was conducted, the assumptions on how the O/Cs might combine their efforts while making assessments (which had been the basis for some of the reports) proved to be incorrect, invalidating some of the requirements for comparison summaries. Last, after completion of the field verification test at JRTC and following an evaluation of the database, one additional report turned out to be of little utility.

As a result of the field verification test, and the data analysis, the task identification levels and two pre-formatted report formats were retained. This database was considered the test database for the process measures and was used as the initial database for the field tryout test at NTC. The list of pre-formatted reports retained for the database and used for the analysis of the data upon completion of the rotation at NTC are shown in Figure 3 below. The detailed requirements, formats, and sample reports in the designated formats are in Appendix A.

Task Identification Levels for Reports
Task Assessment Distribution Summary
Task Remarks Comparison Summary

Figure 3 Pre-formatted Database Reports

In addition to the above requirements, other requirements were identified to allow the user to manipulate the database data using the following variables. A major addition to these was the requirement for a collective, or "All" option to be added to each variable:

Training Center
Rotation
Mission (Attack, Defend, etc.)
Training Day
Echelon (Brigade, Battalion, Task Force, etc.)
Unit
Task List Section (TACP, Maneuver, etc.)
Mission Phase (Planning, Preparation, Execution etc.)
Observer/Controller ID

Figure 4 Required Database Sorting Variables

B. Data Collection

The measurement system designed to collect the data for AGTFS (in both low and mid-high intensity conflicts) consists of both process measures and outcome measures. The instrumented systems provide input to the outcome measures at NTC. The responses to the CAS task lists that have been developed provide the input for the process measures.

At the present time, data for process measures can only be collected manually by observer/controllers (O/Cs) entering comments in paper based forms and booklets or electronic collection instruments. While data for outcome measures can be collected both manually from the O/Cs and electronically through the NTC and AW I instrumentation systems, overall, data collection at the CTCs for CAS is mostly manual.

Data collection for input into the database is designed to follow the After Action Review format. Completion of a database for both the process and outcome measures that receive electronic data input from the CTCs is required so that it will be in place when full instrumentation is installed and working at all the CTCs. This report addresses the development of both the outcome measure and the process measure databases.

Data collection for this study relied on current NTC practices and procedure. In order to evaluate the utility of the database, it was necessary to simulate the type of electronic data input anticipated as a result of future instrumentation improvements. The ECI (electronic collection instrument) equipment and software was utilized to simulate those expected improvements in instrumentation. Data was collected during an NTC Rotation using the ECI format with paper task lists as the field collection system at NTC. The paper based assessments were later entered into the ECI and then downloaded into the database. At AW I, the Forward Air Controllers entered their process measure responses directly into an ECI form which was then downloaded into the database. Copies of task list booklets used for the data collection were included with the field tryout report. A list of the task list booklets used for data collection is included below.

The collection and identification of data to be put into the database is discussed below. The following paragraph on methodology for data collection discusses the way the data was collected and identifies, and provides definitions for, the various forms and assessments that were used. The reports, formats, and displays that were identified as being required from the database are attached at Appendix A.

C. Methodology for Data Collection

The collection and identification of information to be put into the database is discussed below. This section discusses the way the data was collected and defines the various forms and assessments that were used.

The methodology for data collection at NTC was to satellite on the Observer/Controller (O/C) teams at the brigade and task force level. O/Cs act in the capacity of assessors of the unit's employment of close air support as they perform their normal duties of observing and coaching the unit in tactical training. As the unit performed its normal planning, preparation and execution for its mission, the O/Cs observed whether those tasks identified as relating to CAS were accomplished. The O/Cs then made the appropriate assessment entries in booklets.

While the maneuver unit O/Cs (brigade and task force) collected data on the unit's tasks in relation to CAS employment, the Air Force Tactical Air Control Party (TACP) O/Cs collected the data on the unit TACPs and Air Liaison Officers (ALO) for the TACP planning, preparation, and execution of CAS. These O/Cs also made the appropriate assessment entries in booklets.

Air Force Air Forward Air Controller (AFAC) personnel, and contractor personnel, collected the information on the AFAC planning and preparation tasks at AW I. This data was entered directly into the AFAC data forms loaded into the ECI.

In addition, information on Air Control Orders, Air Tasking Orders, sortie allocations, availability of aircraft, preplanned and immediate air requests and Bomb Damage Assessment was collected at the NTC's Training Analysis and Feedback (TAF) Facility and at AW I's flight operations.

D. Process Measures Data Collection Process

The data collection plan described below is the one used during the field tryout test. The data collected during the test was used to fill and test the utility of the prototype database. Some of the procedures and methods were changed as a result of the test. Those changes that affect the design, loading, composition, and use of the database will be discussed later in this report.

The initial CAS task list was divided into seven sections: Maneuver Planning and Maneuver Preparation; TACP Planning and TACP Preparation; AFAC Planning and AFAC Preparation; and CAS Execution. The maneuver tasks had previously been divided into sub-sections for each staff functional area (operations, intelligence, fire support, etc.) for the verification test at JRTC. That was determined to not be a useful step in the evaluation process and was eliminated for the field tryout at NTC. The fire support element was determined to be the maneuver staff element most involved with CAS. When the maneuver tasks were consolidated into one section, the fire support section was retained to provide input to the CAS maneuver tasks. For the NTC verification test, the planning and preparation sections were further combined for each of the functional areas, leaving only four sections (Maneuver, TACP, AFAC, and CAS Execution). Outcome measures were included with the CAS Execution tasks. These sections were then reduced in size, printed, and bound into booklets approximately 5.5" by 8.5" in size so that they would fit in the BDU trousers' cargo pocket. A set of the booklets was included with the report on the verification test (Huffman, 1994). All O/Cs at NTC used the booklets. In addition to the paper collection forms (the booklets), the Electronic Collection Instrument (ECI) was used by AFAC personnel at Air Warrior, Nellis AFB. This provided for both paper and electronic data collection processes to be evaluated. The electronic data was directly entered into the database that was available on site while the verification test was being conducted.

The task list booklets were filled out at both the brigade and the task force echelons (for two task forces) for each mission. Due to shortages of TACP O/Cs, the TACP planning and preparation task lists were filled out twice, during the mid- and end-rotation AARs conducted with the unit ALOs. The AFAC planning and preparation task lists were filled out by the AFACs on each day that missions were flown. The execution task list was filled out by the TACP O/Cs for each CAS mission flown.

Recognizing that different Observer/Controllers (O/Cs) might assess the tasks from any single section, the O/Cs were instructed that any O/Cs entering assessments to a booklet should be identified by call sign. One or more O/Cs could contribute to the booklet for a given mission over the time period for that mission. Several O/Cs could contribute to the ratings on one booklet or data file, either by adding assessments to partially assessed tasks during the same day; or by assessing additional tasks in the same section on different days. O/Cs were instructed to use a separate page for their individual comments. Due to database design and the format of the ECI, it later turned out that only one O/C could be identified with any individual booklet and that it was not possible to separately identify individual O/Cs with separate pages in a book. O/Cs were provided the following scale and meanings for their assessments:

NOT DONE	Unit should have performed the task listed, but did not attempt to perform it.
NOT ADEQUATE	The unit did not perform the task to standard.
MARGINALLY ADEQUATE	The unit successfully performed the task with some shortcomings. The shortcomings were not severe enough to require complete retraining.
ADEQUATE	The unit successfully performed the task to standard. The performance was free of significant shortcomings.
SUPERIOR	The unit exceeded the standard by a significant margin. Unit performance of the task stands out to the extent that it should be cited in the TF AAR. Remarks section should be used to sketch any TTPs that should be relayed to appropriate service schools.
NOT OBSERVED	Unit performed or attempted to perform the task, but OC did not observe the performance or the outcome.
NOT APPLICABLE	The task is not performed by this type or echelon unit. This task is not applicable to the battlefield operating system being evaluated by the OC. The task was not relevant to the given mission.
	Each major task is followed by a remarks section for any additional comments or explanation of the assessment.

Figure 5 Task Assessment Scale

The O/Cs were also given the following instructions on required and optional assessments.

If a major task is assessed "ADEQUATE" or better, then the subordinate tasks need not be addressed unless they are "NOT ADEQUATE".
If a subordinate task is "NOT ADEQUATE", it should be assessed even if the overall task is judged "ADEQUATE" or better.
If a major task is assessed as "NOT ADEQUATE", then all the subordinate tasks need to be individually assessed to identify, in more detail, the rationale for the overall "NOT ADEQUATE" assessment.
If the subordinate tasks do not adequately explain why a major task is being assessed "NOT ADEQUATE", the remarks section should be used to identify the additional factors which caused that assessment.

Figure 6 Required Assessments

Each O/C using the ECI was instructed to select the form that matched the functional area for which the O/C was responsible and the phase of the mission being evaluated. The O/Cs using the booklets were given the appropriate paper forms (booklets) at the beginning of each mission.

After selecting, or being given, the proper form, the O/Cs filled in the header information on the form, which included: Rotation, Training Day, O/C Identity, Unit Mission, CAS Mission, and Unit. This allowed each data collection form to have a unique identifier compiled from the header information on the form.

The booklets, and the forms prepared in the ECI, were given distinctive names and two letter identification codes to facilitate data entry of uniquely identified responses into the database. Each numbered or lettered task and subtask was designed to have a unique task data element field identifier consisting of the one or two letter form prefix identifying the section (see below) and the complete task number. The task numbers for planning and preparation tasks for any given section are numbered consecutively (e.g., TACP Planning section contained tasks G01-G28 and TACP Preparation section contained tasks G29-G35) and those task numbers are discreetly identified in the database with the appropriate phase.

The names and corresponding codes are as follows:

FORM NAME	FORM PREFIX FOR TASK ID
Maneuver [Fire Support] Planning and Preparation	MF
TACP Planning and Preparation	G
AFAC Planning and Preparation	A
CAS Execution and Outcome Measures	GA

Figure 7 Form and Task Identifier Codes

Data collection was conducted using the reduced size, bound booklets containing the required tasks for the maneuver units. The Airborne Forward Air Controller (AFAC) personnel were selected to be trained on the Electronic Collection Instrument (ECI) for use as primary means for the data collection.

Maneuver unit O/C personnel were provided the booklets at the beginning of each of the four force-on-force missions of the rotation and the books were collected during the preparation for the AAR for each mission. In addition, books were also distributed to and collected from the fire support O/C on the Live Fire team for the two live fire missions. Air Force O/Cs and unit ALOs were provided books during the mid-rotation AAR and at the completion of the rotation. Task assessments were made during the AARs and the booklets collected. Execution and outcome measures were assessed by the TACP O/Cs at the Training and Feedback (TAF) facility for each CAS mission flown. All of the assessments at NTC were completed in the books. That data was then entered into the ECI by contractor personnel and downloaded to the database for on site analysis. The AFACs at Nellis completed the AFAC planning and preparation task lists for each of the missions flown and entered their responses directly into the ECI. That data was downloaded to the database by contractor personnel upon completion of the rotation.

The Observer/Controller team and unit player identifiers are shown in the chart below.

Maneuver Observer/Controller Team Position				
	Identifier Call Sign			
	Brigade Team	Heavy TF	Light TF	Live Fire
Fire Support O/Cs	F20	F30	F40	Z27
Air Force TACP Observer/Controller Team and Unit Players				
TACP O/Cs	RAVEN			
Unit ALO/TACPs	ANTI30	TRP60	ANTI32	
AFACs (All)	MISTY			

Figure 8 Observer/Controller Positions and Identifiers.

E. Outcome Measures Data Collection Process

Outcome measures examine the effects of some activity. At the Combat Training Centers, outcome measures are used to exhibit the results of the employment of combat systems on the training battlefield. The results consist of electronically collected force-on-force engagement outcomes, subjective observations of weapons' effects, or an analysis of the participants' actions and their results. Actions are the coordination and communications between participants, tactics, and the synchronization of assets. Specifically, this report is interested with outcome measures for the Air-Ground Training and Feedback System (AGTFS) that capture the effects of close air support aircraft at the NTC.

The outcomes for the AGTFS are the physical actions and effects of close air support at the NTC. Close air support is defined in JCS Pub. 1 as: "Air action against hostile targets which are close to friendly forces and which require detailed integration of each air mission with the fire and movement of those forces."[1] Execution of a close air support mission is the performance of physical actions against hostile targets. The outcome of a close air support mission is the effect of those actions on the target.

[1] Joint Chiefs of Staff, Department of Defense Dictionary of Military and Associated Terms, JCS Publication 1 (Washington, D.C.: U.S. Government Printing Office, 1 January 1986), p. 70.

While the outcomes of a CAS mission are the effects of the weapons on a target, to be useful, outcome measures should allow commanders, trainers, and analysts to understand not only what happened but also why something happened. Outcome measures are more than just a number of weapons launched and number of targets hit. Considerations for the AGTFS outcome measures are shown in Figure 9 below.

Effects of Ground and Air Weapons	• Casualties on the ground caused by air-to-ground weapons. • Aircraft casualties caused by small arms and surface-to-air weapons. • Effects of the air attack on the enemy ground force's actions. • Effects of the air attack on friendly ground forces.
Precision of the Delivery of Air-to-Ground Weapons	• Aircraft attack the correct target. • Weapons hit the correct target.
Tactics Used by Aircraft	• The use of terrain and flight techniques. • The use of countermeasures.
Coordination Between Air and Ground Forces	• Delivery of air-to-ground weapons support the ground mission. • The air attack is synchronized with the ground forces.

Figure 9 Outcome Measure Considerations

The Performance Measure Index (of the ATGFS) is based on the NTC's and Air Warrior I's collective electronic Measurement and Debriefing System (See Performance Measure Index for the Air-Ground Training Feedback System, HumRRO, P. Jarrett, 1994), and is derived from these three measures:

Lethality: The *Lethality Component Measure* (LCM) is a measure of success achieved by the air platforms against ground targets. The LCM measures the performance of all the air crews in a given mission. LCM is not concerned with what target is attacked; it is only concerned with the physical outcome of a given attack. Differentiating between targets is part of the Contribution Component Measure. The LCM consists of a ratio for each mission: The number of kills made divided by the number of air-to-surface weapons used. This measures the effectiveness of the pilots to engage targets with the aircraft's weapon systems. The two components of this measure are collected electronically from the instrumentation system.

Survivability: The *Survivability Component Measure* (SCM) is the ratio of surviving aircraft at the end of a mission to the aircraft available at the beginning of the mission. The SCM is not concerned with why an aircraft is killed, or how it was killed. SCM only measures the percentage of surviving aircraft against total aircraft committed. The SCM is collected and reported in the same manner as the LCM. This measure uses the number of aircraft at the start of a mission and the number of aircraft at the end of a mission to determine the ratio. This is an objective measure collected electronically.

Contribution: The *Contribution Component Measure* (CCM) is a collective measure based upon the plan, coordination, and attack. The CCM is a subjective analysis of how well the ground and air components synchronized the use of CAS. The CCM is collected as a series of observations made by observer/controllers. Observer/controllers make these observations as part of their critique of the elements participating in the training. Observer/controllers gather CCM data at the same time they observe operations. The framework for the analysis lies in the Army's METT-T (mission, enemy, troops, terrain, and time) method for mission analysis:

Mission: The mission is the result of what the ground commander wants to achieve from the close air support sorties. During planning, the ground commander has determined that the combat power of the close air support sorties will accomplish some action that will help the ground forces accomplish their mission. Examples are, the close air support destroys an enemy unit, close air support attacks delay enemy units, or the close air support attacks allow friendly units to maneuver. The mission portion of the CCM is a subjective measure gathered by an Observer Controller. The mission portions will not answer why, but only whether CAS was used as intended.

Enemy: In the LCM the number of enemy vehicles and personnel killed by close air support is collected. In the enemy portion of the CCM O/Cs collect a subjective measure of whether the close air support attacked the correct target. The destruction of many enemy targets by close air support may not have a positive influence on the mission if the correct target is not attacked.

Troops: Fratricide is a continuing problem on the battlefield. As ranges of weapons and mobility of combat vehicles increase, the difficulties involved in correctly identifying enemy and friendly vehicles also increase. This is especially true in a low intensity conflict scenario where friendly and enemy elements may be very close (within small arms range) and interspersed. Procedural controls must be used by the close air support team to eliminate the chances of fratricide; the troops portion of the CCM is a measure of the success of the procedural controls used. The goal of all commanders is zero fratricide.

Terrain: Historically, terrain is used as cover and concealment to protect oneself from enemy fires. With the increasing use of stealth technology, electronics, and other devices, the use of terrain as cover and concealment for CAS aircraft may not be as necessary on the modern battlefield. Yet, the close air support aircraft must take measures to protect themselves from enemy and friendly fires. The terrain portion of the CCM is a subjective observation of whether the close air support aircraft used the correct flight tactics, air defense countermeasures, and airspace control measures.

Time: Time is a critical function of synchronizing combat forces on the battlefield. Close air support aircraft arriving too early may run out of station time prior to the having an opportunity to attack; aircraft arriving late may miss the opportunity to synchronize their attack with other combat forces. Time is also a function of how well the ground commander has foreseen the events on the battlefield. The time portion of the CCM is not concerned with what went wrong or why someone was early or late, only if they were. This factor is not only concerned with whether the aircraft was on time, but also must address how the commander synchronizes aircraft that become available when he has not planned for them. Time, in this case, is a subjective judgement by the observer/controller of the commander synchronizing the close air support into the battle with his other fire support assets.

Collection of the CCM has to be accomplished from several different locations by observer/controllers. The Mission, Enemy and Time portions must be collected by someone who is familiar with the what the commander intended his close air support to accomplish. This observer/controller will have the be positioned with the headquarters that is planning and coordinating the attack. Collection of the Troops and Terrain portions can be made by observer/controllers who can observe the target and attacking aircraft. This data can be collected at the same time as the LCM and the SCM.

Observer/controllers answer the specific questions below in conducting the METT-T analysis of a CAS mission:

Mission:	Was the mission assigned by the TF commander to the CAS accomplished?
Enemy:	Was the correct enemy force, or engagement area attacked?
Terrain:	Did the CAS aircraft use the proper tactics and/or counter measures during their attack?
Troops:	Were friendly forces attacked by the CAS aircraft or were friendly aircraft engaged by friendly ground fires?
Time:	Did the CAS aircraft attack within the time window designated by the ground commander, or did the ground commander synchronize the CAS into the battle?

The three measures (lethality, survivability, and contribution) are collected by both electronic means and observer/controller comments. The measures are then available for use as a collective measure or as individual measures for use in after action reviews at the NTC and Air Warrior I or as data for analysis.

The LCM, SCM, and CCM do not answer why something was successful or was not successful. They are the basis for developing empirical relationships among various observations about employment of close air support. The observer/controllers must analyze this, and other, data to determine why a particular battle had a particular outcome. The commander determines the degree of success of that outcome, and of the training.

After collection, the observations must be combined into one package for each mission. Ultimately, this will entail facsimile or electronic transmission of the collected data and observations to a central location. Once all of the data and observations have been delivered to a central location, they can be assembled into one package for debriefings and analysis.

F. Database Specifications

The CAS databases were constructed using FoxPro for Windows V 2.5, but any XBASE compatible database manager should be able to read in and enable the user to manipulate or view the database data.

Initial, or prototype, databases were constructed during the field verification test of the close air support battle task measurement system at JRTC. Data was collected manually at JRTC and was entered into the database upon return to ARI POM. This prototype CAS database, with the six different maneuver databases was then used to analyze the results and to evaluate the applicability of the tasks and subtasks. As a result of that analysis, the requirements were modified, and the database structure was changed to reflect the evolving requirements. The revised database was the starting point for the next phase of the two close air support projects, which was the field tryout of the close air support performance measurement system at NTC. There was no outcome database developed prior to the JRTC field tryout. Upon completion of the analysis of the outcome data collection process at JRTC and NTC, tentative report formats and collection formats were developed. These were then used to collect outcome data during the field tryout test at NTC.

The prototype database was used to compile and analyze the data derived from the field tryout test at NTC. That analysis identified modifications that were required for the final CAS database. The major changes to the prototype database were a reordering of the performance tasks (process measures), elimination of several tasks, deletion of some variables and/or report selection criteria from the database, deletion of one pre-set report, finalization of the outcome measures report formats, and the addition of Maneuver Execution tasks.

The prototype CAS database has been retained with the data from the two field tryout tests. The final CAS database programs were developed and the database structure, with its reports, was established. That database, however, will contain no data until further data is collected at the training centers using the final task list and outcome measures. The resulting databases, files, and reports for the final CAS database are described in the following paragraphs and the programs and file summaries are included as appendices.

The CAS database employs a Graphical User Interface (GUI) that enables the user to enter data into the Master and Outcome databases. There are preset reports available from the master database. The program allows the user to establish the criteria, or variables, for a query from the master database using drop-down list menus.

The user executes the CAS Project Application (by selecting "casproj.app") to start the set of CAS programs. The CAS Project Application "modifies" the FoxPro root menu at the top of the screen. By selecting the "Close Air Support" title from the title bar, the user is presented with a drop-down menu similar to the following:

Close Air Support

- Data Entry - CAS Master Database
- Data Entry - CAS Outcome Database
- Report - CAS Master Database
- Report - CAS Outcome Database
- ECI - CAS Conversion

Selection of the "Data Entry" menu item for the CAS Master Database prompts a sub-menu which allows the user to identify the type of tasks for which data is to be entered. That menu allows selection of the categories of tasks shown below.

- AFAC Planning
- AFAC Preparation
- MANEUVER Planning
- MANEUVER Preparation
- MANEUVER Execution
- TACP Planning
- TACP Preparation
- CAS Execution

Selection of the "Data Entry" menu item for the CAS Outcome Database takes the user directly to a blank form into which outcome data can be entered. Both the master and the outcome database data entry formats first require header information which uniquely identifies the information being entered. This header information includes: Rotation, Training Day, O/C Identity, and Mission. Sample outcome reports are included in Appendix B.

Selection of the "ECI" menu item takes the user to a directory for identification of the ECI database to be converted to the CAS database format and incorporated into the CAS Master Database. The ECI conversion program guides the user through a series of functions to complete the conversion and transfer the ECI data to the CAS database.

To facilitate research and analysis of the task data collected during the field tryout, the data was organized to allow access through a menu selection format. Programs were written to allow the use of the mission variables (previously identified) to build the pre-formatted reports. Selection of the "Report" menu item prompts the compilation of data into the database program for selection of report criteria. Once the data has been compiled, a report menu is displayed that allows selection of options to build the specific report desired. It is possible to select reports for both the process measures ("Report - CAS Master Database") and the outcome measures ("Report - CAS Outcome Database"). The report selection menu for the master database allows for the identification of five report criteria, each of which has several options. The process report and query selections that result from selecting "Report - CAS Master Database" are shown below.

<u>REPORT SELECTIONS</u>	<u>QUERY SELECTION</u>
Title Selection	Rotation Number
Summary Selection	Mission
Task Type Selection	Training Day
Task Phase Selection	Unit Observed
Echelon Level Selection	O/C
	Report (Task) Level

The five report selections allow the user to identify the type and scope of report desired. <u>Title selection</u> allows identification of one of two pre-formatted reports. The first shows the distribution of the task assessments (adequate, not adequate, marginal, etc.) for each task or subtask. The second is a summary of the O/C remarks made, organized by major task. The <u>summary selection</u> identifies the time period desired (mission, day, rotation, all rotations at a training center). <u>Task type selection</u> identifies the four major types or categories of CAS tasks, primarily organized by the organization performing the tasks. <u>Task phase selection</u> identifies the phase of the operation. <u>Echelon level selection</u> allows identification of the army organization level desired. The various options for the report selections are shown in Table 1, on the following page.

REPORT SELECTIONS	**OPTIONS**
Title Selection	Task Assessment Distribution Task Remarks Comparison
Summary Selection	Training Day Mission Rotation Training Center
Task Type Selection	AFAC TACP Maneuver CAS Execution
Task Phase Selection	All Plan Prepare Execution
Echelon Level Selection	All Battalion Task Force Brigade Division Corps Company

Table 1 Report selections and all associated options.

The report criteria are further defined, or narrowed, by selecting from an additional six query options. The Rotation Number is the four digit, alpha-numeric code for all rotations from which data has been collected. Mission lists all the maneuver unit missions conducted during the observed rotations. The Training Day selection lists the specific days in the rotation that missions were conducted, identified by a number or letter and number. The Unit Observed selection lists the unit designation of each unit for which data has been collected during a rotation. O/C lists all the Observer/Controller call signs for those O/Cs who have provided assessments and comments. The Report Level selection allows the user to identify the task/sub-task level desired for the task assessments in the designated report. The query selections available, and a sample content of the different options for each, are listed in Table 2, on the following page.

QUERY SELECTION	**MENU OPTIONS**
Rotation Number	All J945 N949
Mission	All MTC ATK FORCED ENTRY AIR ASLT
Training Day	All 1 5 D-Day D-1 D+2
Unit Observed	All 2BDE, __DIV 3BDE, __DIV
O/C	All Y03 S05
Report (Task) Level	Level 1 (M01) Level 2 (M01a) Level 3 (M01a1) Level 4 (All task numbering)

Table 2 Query Selection and sample associated options

The outcome report and query selections resulting from selecting "Report - CAS Outcome Database" are shown below. This menu allows the user to select one of the four listed outcome report summaries and specify report content using the query selection criteria.

<u>OUTCOME REPORTS</u> - Select	<u>QUERY SELECTION</u>
Rotation Summary	Rotation Number
Mission Summary	Mission
Day Summary	Training Day
Comments Summary	

The various options available for the query selections are shown in Table 3, below. The Rotation Number is the four digit, alpha-numeric code for all rotations from which data has been collected. Mission lists all the maneuver unit missions conducted during the observed rotations. The Training Day selection lists the specific days in the rotation that missions were conducted, identified by a number or letter and number. Sample outcome reports are at Appendix B.

QUERY SELECTION	**MENU OPTIONS**
Rotation Number	All J945 N949
Mission	All MTC DEFENSE FORCED ENTRY AIR ASLT
Training Day	All 1 5 D-Day D-1 D+2

Table 3 Query Selection and sample associated options.

VII. DATABASE MANAGEMENT FILES AND PROGRAMS

A. CAS Database System Summary

The Close Air Support database contains a series of interacting programs, procedures, functions, databases, tables, report forms, and menus. The system has 8039 lines of code which are contained in two project files, thirteen program and procedure files, 38 procedures and functions, 24 databases and tables, seven report forms and a menu file. The following paragraphs describe the various files and programs that were developed to manage the CAS database and provide for the preset report generation. At Appendix C is a tree diagram that depicts the interaction and relationships of the different programs, procedures, and functions. Appendix D is a list of the files in the CAS database.

B. CAS Database Project Files

The overall CAS set of programs, reports, menus, and databases are managed via a project file called "casproj.pjx". This file enables the FoxPro user to group program files and databases together for easier project management. The project program also combines all of the programs, reports, and menus into an application program. The overall application program for the CAS databases is "casproj.app".

C. CAS Database Menu Files

CAS user selection menu for data entry, reports, or conversion. The menu file summary (casmenu.mnx) is at Appendix E. The menu program with definitions and procedures (casmenu.mpr) is at Appendix I.

D. CAS Database Procedures and Function Files

There are thirty-eight procedures and functions in the CAS database. Appendix F lists the procedures and functions sorted by the programs in which they are used.

E. CAS Database Procedure and Program Files

The names and descriptions of the thirteen procedure and program files developed for this database are listed below. A summary of the seven programs that contain procedures and functions is included in Appendix F. The text of the program files are attached as Appendixes I-T.

casmenu.mpr	CAS menu definition and procedure file (Appendix I)
casentry.prg	Enter data into the casdata.dbf database (Appendix J)
casout1.prg	Enter data into the casoutc.dbf database (Appendix K)
casrpts.prg	User interface to SQL query and reports (Appendix L)
caserr.prg	Inform the user of program/application errors (Appendix M)
cascrpt1.prg	SQL queries for the Task Assessment Consistency Report (Appendix N)
cascrpt2.prg	SQL queries for the Task Assessment distribution Report (Appendix O)
cascrpt3.prg	SQL queries for the Task Remarks Comparison Report (Appendix P)
casnone.prg	Inform the user that no data was found per his query request (Appendix Q)
casorpts.prg	User interface to SQL query and outcome reports (Appendix R)
casorot.prg	SQL queries for the Outcome Reports (Appendix S)
casconv.prg	Convert ECI data to the casdata or casoutc database format (Appendix T)
casloc.prg	Select option to print report to printer or to a file (Appendix U)

Figure 10 CAS Procedure and Program File Names

F. CAS Database Files and Tables

There are twenty-four database files and tables in the CAS database. There are two primary, variable, CAS databases. One accumulates the task number accomplishment levels (process data) and the other is used to record lethality, survivability, and contribution data (outcome data). The fields contained in the two primary databases are described in tables 4 and 5 below. In addition, there are twenty-two supplemental, static, databases that are used in conjunction with the CAS Master Database and CAS Outcome Database to formulate relational database SQL queries. The files names and a brief description for each are listed in Appendix G.

CAS Master Database (casdata.dbf)			
Field Name	Field Type	Field Size	Contains
rotation	char	4	CTC Rotation identification
trng_day	char	8	Training day
time	char	4	Time mission observed
unit_obs	char	15	Designation of unit observed
oc_cs	char	15	Designation of O/C commenting
mission	char	15	Type mission
cas_mis	char	4	CAS mission number
task_id	char	2	Two letter task section identification

Table 4 CAS Master Database Fields

CAS Outcome Database (casoutc.dbf)			
Field Name	Field Type	Field Size	Contains
rotation	char	4	CTC Rotation identification
mission	char	15	Type mission
oc_cs	char	4	Designation of O/C commenting
dtg	char	15	Date-time group of mission observed
leth_a	numeric	2	Number of weapons used
leth_b	numeric	2	Number of vehicles killed
surv_a	numeric	2	Number of aircraft starting mission
surv_b	numeric	2	Number of aircraft at end of mission
com_mis	numeric	1	Yes/no response to Mission question
com_ene	numeric	1	Yes/no response to Enemy question
com_tro	numeric	1	Yes/no response to Troops question
com_ter	numeric	1	Yes/no response to Terrain question
com_tim	numeric	1	Yes/no response to Time question
rem_mis	memo	10	Narrative remarks on Mission
rem_ene	memo	10	Narrative remarks on Enemy
rem_tro	memo	10	Narrative remarks on Troops
rem_ter	memo	10	Narrative remarks on Terrain
rem_tim	memo	10	Narrative remarks on Time

Table 5 CAS Outcome Database Fields

G. Report Form Files

There are seven report form files in the CAS database. Three of these are process reports and four are outcome reports. The list of report form files is at Appendix H. The programs for generating these reports are at Appendices L, N, O, P, R, and S.

H. Collective Air Ground Training and Feedback System Database

A copy of the AGTFS database on disc is attached as Appendix V. This is the shell database with all programs, files, and databases, but without the data from the two test rotations.

APPENDIX A

DATABASE REPORT REQUIREMENTS AND SAMPLE FORMATS

This appendix describes the database report requirements as they were identified for the initial database development.

1. The required database reports (listed below) all need to be available at different levels of detail with respect to the task identification. The first level is tasks identified at the major task level [ie. MO18]. The second level is tasks identified to the sub-task level [ie. MO18a]. The third level is tasks identified to the lowest level available in the database [ie. MO18b1)]. The example reports in the next section are all shown at the first level, or only to the major task level.

    a. An example of the task ID listing for the three levels is:

LEVEL 1	LEVEL 2	LEVEL 3
TASK ID	TASK ID	TASK ID
MO18 [TEXT]	MO18 [TEXT]	MO18 [TEXT]
	MO18a [TEXT]	MO18a [TEXT]
	MO18b [TEXT]	MO18a1) [TEXT]
		MO18a2) [TEXT]
		MO18b [TEXT]
		MO18b1) [TEXT]
		MO18b2) [TEXT]
		MO18b3) [TEXT]

Chart: Task Identification Levels for Reports

    b. The user or report requester needs to be able to specify the level of task identification desired or required.

    c. In the numbered sections below, the example reports show only the task identification numbers. In the generated reports, the task number and the text of the task identified by the number are both required.

2. List, by task for each task list section: by training day, by mission, by rotation and/or by training center; and by unit, by echelon:

   a. The count of each assessment category (not done, not adequate, marginally adequate, adequate, superior, not observed, and/or not applicable; and given no score) for each task.

   b. Intent is to show the range and distribution of scores for each task for each task list section; unit, echelon; training day, mission, rotation and/or training center.

GENERIC REPORT TITLE: Items shown in brackets [...] are the different possible variables that can be selected to identify the particular data set being summarized for a report.

TASK ASSESSMENT DISTRIBUTION, [training day, mission, rotation, training center] SUMMARY
[task section] TASKS, [echelon] LEVEL
[Identifier information - specific rotation number;
unit identification;
mission IDs of missions included in the summary.]

   c. EXAMPLE:

TASK ASSESSMENT DISTRIBUTION, ROTATION SUMMARY
MANEUVER [OPERATIONS] PLANNING TASKS, BRIGADE LEVEL
ROTATION 94-5; 3D BRIGADE, 82D ABN; MISSIONS 45-1, 45-2, 45-3, 45-4, 45-5

TASK ID	NOT DONE	NOT ADQ	MARG ADQ	ADQ	SUP	NOT OBS	NOT APP	NO ASSESS
MO01		2	1	1		1		
MO02	1	1	1	2				
MO03		1	1	3				
MO04							5	
MO05							5	

NOTE: *Task titles in text should follow each task ID number in this report and all following reports.*

3. List, by task for each task list section: by training day, by mission, by rotation and/or by training center; and by unit, by echelon:

a. The text of the remarks made for each task for the period of the report.

b. Intent is to list all the comments or remarks made about each task in a task list section together for comparison. The source of each remark must be identified (day, mission, O/C, etc.)

GENERIC REPORT TITLE: Items shown in brackets [...] are the different possible variables that can be selected to identify the particular data set being summarized for a report.

TASK REMARKS COMPARISON, [training day, mission, rotation, training center] SUMMARY
[task section] TASKS, [echelon] LEVEL
[Identifier information - specific rotation number; unit identification; mission IDs of missions included in the summary.]

c. EXAMPLE:

TASK REMARKS COMPARISON, ROTATION SUMMARY,
MANEUVER [OPERATIONS] PLANNING TASKS, BRIGADE LEVEL
ROTATION 94-5; 3D BRIGADE, 82D ABN; MISSIONS 45-1, 45-2, 45-3, 45-4, 45-5

MANEUVER [OPERATIONS] PLANNING		
MO01		
	MISSION 45-1: O/C 1:	[TEXT]
	MISSION 45-4: O/C 2:	[TEXT]
MO02	MISSION 45-3: O/C 1:	[TEXT]
	MISSION 45-4 O/C 1:	[TEXT]

APPENDIX B

SAMPLE OUTCOME REPORTS

This appendix contains four sample outcome reports. The preset report format titles and the data included in each report are described below.

CAS Outcome Rotation Summary:
: Summarizes the lethality and survivability data for all missions flown during the entire rotation. Shows the total number of "Yes" and "No" responses to the contribution component questions.

CAS Outcome Mission Summary:
: Summarizes the lethality and survivability data for all CAS missions flown during a specific ground maneuver mission. Shows the total number of "Yes" and "No" responses to the contribution component questions and a summary of the comments made in addition to the "Yes/No" responses.

CAS Outcome Day Summary:
: Summarizes the lethality and survivability data for all missions flown during an entire day. Shows the total number of "Yes" and "No" responses to the contribution component questions. This is the same format as the rotation summary, but for a shorter time period.

CAS Outcome Comments, Rotation Summary:
: Summarizes the comments made in response to the contribution component questions. Report produces a separate page of comments for each mission flown which resulted in outcomes assessed.

CAS Outcome Rotation Summary

Rotation:	All	**# of Air Missions:**	9	**Training Day:**	All

Leathality Component

A. Total Number of Weapons Used:	101
B. Total Number of Vehicles Killed:	33

Survivability Component

A. Total Number of Aircraft Starting Mission:	28
B. Total Number of Aircraft at End of All Missions:	18

Contribution Component

MISSION:	Did the CAS mission accomplish the task assigned by the ground commander?	Yes	6	No	3
ENEMY:	Was the correct enemy force or engagement area attacked?	Yes	7	No	2
TROOPS:	Were friendly forces attacked by the CAS or the friendly aircraft destroyed by friendly ADA or ground fires?	Yes	1	No	8
TERRAIN:	Did the CAS aircraft use the proper tactics or counter measures during their attack?	Yes	7	No	2
TIME:	Did the CAS aircraft attack within the time windows designated by the ground commander, or did the ground commander synchronize the CAS into the battle?	Yes	5	No	4

CAS Outcome Mission Summary

Rotation: All **Air Mission #:** MTC **Training Day:** All

Leathality Component

A. Total Number of Weapons Used:	28
B. Total Number of Vehicles Killed:	7

Survivability Component

A. Total Number of Aircraft Starting Mission:	6
B. Total Number of Aircraft at End of All Missions:	2

Contribution Component and Comments

MISSION: Did the CAS mission accomplish the task assigned by the ground commander?

Yes 1 NO ASSIGNED/SYNCHRONIZED BDE PLAN.

No 1

ENEMY: Was the correct enemy force or engagement area attacked?

Yes 2

No 0

TROOPS: Were friendly forces attacked by the CAS or the friendly aircraft destroyed by friendly ADA or ground fires?

Yes 0 BUT POSSIBLE DUE TO MIXED FORCES.

No 2

TERRAIN: Did the CAS aircraft use the proper tactics or counter measures during their attack?

Yes 2

No 0

TIME: Did the CAS aircraft attack within the time windows designated by the ground commander, or did the ground commander synchronize the CAS into the battle?

Yes 0

No 2

CAS Outcome Day Summary

Rotation: All **# of Air Missions:** 2 **Training Day:** 14

Leathality Component

A. Total Number of Weapons Used:	28
B. Total Number of Vehicles Killed:	9

Survivability Component

A. Total Number of Aircraft Starting Mission:	8
B. Total Number of Aircraft at End of All Missions:	4

Contribution Component

MISSION: Did the CAS mission accomplish the task assigned by the ground commander?

Yes 2 No 0

ENEMY: Was the correct enemy force or engagement area attacked?

Yes 2 No 0

TROOPS: Were friendly forces attacked by the CAS or the friendly aircraft destroyed by friendly ADA or ground fires?

Yes 1 No 1

TERRAIN: Did the CAS aircraft use the proper tactics or counter measures during their attack?

Yes 1 No 1

TIME: Did the CAS aircraft attack within the time windows designated by the ground commander, or did the ground commander synchronize the CAS into the battle?

Yes 2 No 0

CAS Outcome Comments, Rotation Summary

Rotation: N949 **Air Mission #:** MTC **Training Day:** 1 **OC C/S:** RAVEN

Contribution Component and Comments

MISSION: Did the CAS mission accomplish the task assigned by the ground commander?

NO ASSIGNED/SYNCHRONIZED BDE PLAN.

ENEMY: Was the correct enemy force or engagement area attacked?

TROOPS: Were friendly forces attacked by the CAS or the friendly aircraft destroyed by friendly ADA or ground fires?

BUT POSSIBLE DUE TO MIXED FORCES.

TERRAIN: Did the CAS aircraft use the proper tactics or counter measures during their attack?

TIME: Did the CAS aircraft attack within the time windows designated by the ground commander, or did the ground commander synchronize the CAS into the battle?

CAS Outcome Comments, Rotation Summary

Rotation: N949 **Air Mission #:** SORTIE 2 **Training Day:** 1 **OC C/S:** RAVEN

Contribution Component and Comments

MISSION: Did the CAS mission accomplish the task assigned by the ground commander?

ENEMY: Was the correct enemy force or engagement area attacked?

AFAC GRID FOR CAS TGT WAS 10 K OFF.

TROOPS: Were friendly forces attacked by the CAS or the friendly aircraft destroyed by friendly ADA or ground fires?

UNCONFIRMED REPORT OF ONE AIRCRAFT ATTACKING A VEHICLE SOUTH OF EA. THAT LOCATION HAD A BMP & M2 NEXT TO EACH OTHER.

TERRAIN: Did the CAS aircraft use the proper tactics or counter measures during their attack?

TIME: Did the CAS aircraft attack within the time windows designated by the ground commander, or did the ground commander synchronize the CAS into the battle?

CAS NOT SYNCHRONIZED W/BDE PLAN.

CAS Outcome Comments, Rotation Summary

Rotation: N949 **Air Mission #:** SORTIE 1 **Training Day:** 3 **OC C/S:** RAVEN

Contribution Component and Comments

MISSION: Did the CAS mission accomplish the task assigned by the ground commander?

MARGINAL

ENEMY: Was the correct enemy force or engagement area attacked?

TROOPS: Were friendly forces attacked by the CAS or the friendly aircraft destroyed by friendly ADA or ground fires?

TERRAIN: Did the CAS aircraft use the proper tactics or counter measures during their attack?

TIME: Did the CAS aircraft attack within the time windows designated by the ground commander, or did the ground commander synchronize the CAS into the battle?

CAS Outcome Comments, Rotation Summary

Rotation: N949 **Air Mission #:** SORTIE 2 **Training Day:** 3 **OC C/S:** RAVEN

Contribution Component and Comments

MISSION: Did the CAS mission accomplish the task assigned by the ground commander?

DIFFICULT TO SAY DUE TO UNREALISTIC BDA ASSESSENTS BY NTC OCS.

ENEMY: Was the correct enemy force or engagement area attacked?

TROOPS: Were friendly forces attacked by the CAS or the friendly aircraft destroyed by friendly ADA or ground fires?

TERRAIN: Did the CAS aircraft use the proper tactics or counter measures during their attack?

TENDENCY TO DO RE-ATTACKS WITHOUT DEPARTING TGT AREA AND BREAK LINE OF SIGHT AND COME IN FROM DIFFERENT DIRECTION.

TIME: Did the CAS aircraft attack within the time windows designated by the ground commander, or did the ground commander synchronize the CAS into the battle?

CAS Outcome Comments, Rotation Summary

Rotation: N949 **Air Mission #:** SORTIE 3 **Training Day:** 3 **OC C/S:** RAVEN

Contribution Component and Comments

MISSION: Did the CAS mission accomplish the task assigned by the ground commander?

BIG TIME.

ENEMY: Was the correct enemy force or engagement area attacked?

TROOPS: Were friendly forces attacked by the CAS or the friendly aircraft destroyed by friendly ADA or ground fires?

TERRAIN: Did the CAS aircraft use the proper tactics or counter measures during their attack?

TIME: Did the CAS aircraft attack within the time windows designated by the ground commander, or did the ground commander synchronize the CAS into the battle?

CAS Outcome Comments, Rotation Summary

Rotation: N949 **Air Mission #:** MTC **Training Day:** 12 **OC C/S:** RAVEN

Contribution Component and Comments

MISSION: Did the CAS mission accomplish the task assigned by the ground commander?

ENEMY: Was the correct enemy force or engagement area attacked?

TROOPS: Were friendly forces attacked by the CAS or the friendly aircraft destroyed by friendly ADA or ground fires?

TERRAIN: Did the CAS aircraft use the proper tactics or counter measures during their attack?

TIME: Did the CAS aircraft attack within the time windows designated by the ground commander, or did the ground commander synchronize the CAS into the battle?

CAS SHOWED UP 30 MIN LATE AND NEVER GOT INTEGRATED INTO BATTLE.

CAS Outcome Comments, Rotation Summary

Rotation: N949 **Air Mission #:** SORTIE 1 **Training Day:** 14 **OC C/S:** RAVEN

Contribution Component and Comments

MISSION: Did the CAS mission accomplish the task assigned by the ground commander?

ENEMY: Was the correct enemy force or engagement area attacked?

TROOPS: Were friendly forces attacked by the CAS or the friendly aircraft destroyed by friendly ADA or ground fires?

TERRAIN: Did the CAS aircraft use the proper tactics or counter measures during their attack?

TIME: Did the CAS aircraft attack within the time windows designated by the ground commander, or did the ground commander synchronize the CAS into the battle?

CAS Outcome Comments, Rotation Summary

Rotation: N949 **Air Mission #:** SORTIE 2 **Training Day:** 14 **OC C/S:** RAVEN

Contribution Component and Comments

MISSION: Did the CAS mission accomplish the task assigned by the ground commander?

ENEMY: Was the correct enemy force or engagement area attacked?

TROOPS: Were friendly forces attacked by the CAS or the friendly aircraft destroyed by friendly ADA or ground fires?

2 FRATS BY SPAD 31

TERRAIN: Did the CAS aircraft use the proper tactics or counter measures during their attack?

SPAD 31 CIRCLED THE TARGET AREA DID 6 REATTACKS , NEVER LEFT TGT AREA.

TIME: Did the CAS aircraft attack within the time windows designated by the ground commander, or did the ground commander synchronize the CAS into the battle?

CAS Outcome Comments, Rotation Summary

Rotation: **Air Mission #:** **Training Day:** **OC C/S:**

Contribution Component and Comments

MISSION: Did the CAS mission accomplish the task assigned by the ground commander?

ENEMY: Was the correct enemy force or engagement area attacked?

TROOPS: Were friendly forces attacked by the CAS or the friendly aircraft destroyed by friendly ADA or ground fires?

TERRAIN: Did the CAS aircraft use the proper tactics or counter measures during their attack?

TIME: Did the CAS aircraft attack within the time windows designated by the ground commander, or did the ground commander synchronize the CAS into the battle?

APPENDIX C

CLOSE AIR SUPPORT DATABASE FUNCTIONAL TREE DIAGRAM

1. System: Close Air Support

2. Author: Dave Butterfield/Jerry Fargo

3. The Tree Diagram depicts the structure and relationships among the various programs, functions, and procedures used in this database. The different programs, functions, and procedures are described in greater detail in subsequent appendices.

```
CASMENU.MPR
├──────_QU70J2FB9     (procedure in CASMENU.MPR)
│   └──────CASOUT1.PRG
│          └──────_WIN_LOWER()  (function in CASENTRY.PRG)
├──────_QU70J2FCT     (procedure in CASMENU.MPR)
│   └──────CASRPTS.PRG
│          ├──────CASERR.PRG
│          ├──────_CHKFORUPDATE     (procedure in CASRPTS.PRG)
│          ├──────_WAITMSG     (procedure in CASRPTS.PRG)
│          ├──────_ROTATION     (procedure in CASRPTS.PRG)
│          ├──────_MISSION     (procedure in CASRPTS.PRG)
│          ├──────_TRAINING     (procedure in CASRPTS.PRG)
│          ├──────_CALLSIGN     (procedure in CASRPTS.PRG)
│          ├──────_UNITOBS     (procedure in CASRPTS.PRG)
│          ├──────_CREATE_RPT()  (function in CASRPTS.PRG)
│          │      ├──────_GET_RPT     (procedure in CASRPTS.PRG)
│          │      ├──────_GET_BOOK     (procedure in CASRPTS.PRG)
│          │      ├──────_GET_TYPE     (procedure in CASRPTS.PRG)
│          │      ├──────_GET_ECHLEV     (procedure in CASRPTS.PRG)
│          │      ├──────CASCRPT1.PRG
│          │      ├──────CASNONE.PRG
│          │      ├──────CASLOC.PRG
│          │      ├──────CASCRPT2.PRG
│          │      └──────CASCRPT3.PRG
│          └──────_STOP_LOOP()  (function in CASRPTS.PRG)
│
```

```
├──────_QU70J2FEC     (procedure in CASMENU.MPR)
│   └──────CASORPTS.PRG
│          ├──────_MAKE_ARRAYS     (procedure in CASORPTS.PRG)
│          ├──────_DOREPORTS()  (function in CASORPTS.PRG)
│          │   └──────CASOROT.PRG
│          │          ├──────_ROT_SUMMARY     (procedure in CASOROT.PRG)
│          │          │   └──────CASLOC.PRG...
│          │          ├──────_MIS_SUMMARY     (procedure in CASOROT.PRG)
│          │          │   └──────CASLOC.PRG...
│          │          ├──────_DAY_SUMMARY     (procedure in CASOROT.PRG)
│          │          │   └──────CASLOC.PRG...
│          │          ├──────_CMT_SUMMARY     (procedure in CASOROT.PRG)
│          │          │   └──────CASLOC.PRG...
│          │          └──────CASNONE.PRG...
│          └──────_WIN_LOWER() ... (function in CASENTRY.PRG)
├──────_QU70J2FLO     (procedure in CASMENU.MPR)
│   └──────CASENTRY.PRG
│          ├──────TMPARRAY()  (function in ?)
│          └──────_WIN_LOWER() ... (function in CASENTRY.PRG)
├──────_QU70J2FNM     (procedure in CASMENU.MPR)
│   └──────CASENTRY.PRG...
├──────_QU70J2FPI     (procedure in CASMENU.MPR)
│   └──────CASENTRY.PRG...
├──────_QU70J2FR2     (procedure in CASMENU.MPR)
│   └──────CASENTRY.PRG...
├──────_QU70J2FSN     (procedure in CASMENU.MPR)
│   └──────CASENTRY.PRG...
├──────_QU70J2FUJ     (procedure in CASMENU.MPR)
│   └──────CASENTRY.PRG...
├──────_QU70J2FWF     (procedure in CASMENU.MPR)
│   └──────CASENTRY.PRG...
├──────_QU70J2FYA     (procedure in CASMENU.MPR)
│   └──────CASENTRY.PRG...
├──────_QU70J2G35     (procedure in CASMENU.MPR)
│   └──────CASCONV.PRG
│          ├──────CASERR.PRG...
│          ├──────ECI_ARRAY()  (function in ?)
│          ├──────_WAITWINDOW     (procedure in CASCONV.PRG)
│          ├──────_GET_TASKID     (procedure in CASCONV.PRG)
│          ├──────_CHK_TASKNO     (procedure in CASCONV.PRG)
│          └──────_PUT_FIELDS     (procedure in CASCONV.PRG)
└──────_QU70J2G4O     (procedure in CASMENU.MPR)
    └──────CASCONV.PRG...
```

APPENDIX D

CLOSE AIR SUPPORT DATABASE FILE LIST

1. System: Close Air Support

2. Author: Dave Butterfield/Jerry Fargo

3. This appendix is a list of the files used for the Close Air Support database. The list is divided into five functional areas: Menu, Programs, Procedures, Databases/Tables, and Reports.

----

1. Menu files:

   CASMENU.MNX

2. Program files:

   CASMENU.MPR
   CASENTRY.PRG
   CASRPTS.PRG
   CASCONV.PRG
   CASOUT1.PRG
   CASOROT.PRG
   CASORPTS.PRG

3. Procedure files:

_QU70J2FB9	(procedure in CASMENU.MPR)
_QU70J2FCT	(procedure in CASMENU.MPR)
_QU70J2FEC	(procedure in CASMENU.MPR)
_QU70J2FLO	(procedure in CASMENU.MPR)
_QU70J2FNM	(procedure in CASMENU.MPR)
_QU70J2FPI	(procedure in CASMENU.MPR)
_QU70J2FR2	(procedure in CASMENU.MPR)
_QU70J2FSN	(procedure in CASMENU.MPR)
_QU70J2FUJ	(procedure in CASMENU.MPR)
_QU70J2FWF	(procedure in CASMENU.MPR)
_QU70J2FYA	(procedure in CASMENU.MPR)

_QU70J2G35	(procedure in CASMENU.MPR)
_QU70J2G4O	(procedure in CASMENU.MPR)
TMPARRAY()	(function in ?)
_WIN_LOWER()	(function in CASENTRY.PRG)
_CHKFORUPDATE	(procedure in CASRPTS.PRG)
_WAITMSG	(procedure in CASRPTS.PRG)
_ROTATION	(procedure in CASRPTS.PRG)
_MISSION	(procedure in CASRPTS.PRG)
_TRAINING	(procedure in CASRPTS.PRG)
_CALLSIGN	(procedure in CASRPTS.PRG)
_UNITOBS	(procedure in CASRPTS.PRG)
_CREATE_RPT()	(function in CASRPTS.PRG)
_STOP_LOOP()	(function in CASRPTS.PRG)
_GET_RPT	(procedure in CASRPTS.PRG)
_GET_BOOK	(procedure in CASRPTS.PRG)
_GET_TYPE	(procedure in CASRPTS.PRG)
_GET_ECHLEV	(procedure in CASRPTS.PRG)
ECI_ARRAY()	(function in ?)
_WAITWINDOW	(procedure in CASCONV.PRG)
_GET_TASKID	(procedure in CASCONV.PRG)
_CHK_TASKNO	(procedure in CASCONV.PRG)
_PUT_FIELDS	(procedure in CASCONV.PRG)
_INVALID_DATA	(procedure in CASOUT1.PRG)
_ROT_SUMMARY	(procedure in CASOROT.PRG)
_MIS_SUMMARY	(procedure in CASOROT.PRG)
_DAY_SUMMARY	(procedure in CASOROT.PRG)
_CMT_SUMMARY	(procedure in CASOROT.PRG)
_MAKE_ARRAYS	(procedure in CASORPTS.PRG)
_DOREPORTS()	(function in CASORPTS.PRG)

4. Tables/databases:

CASABK.DBF
CASDATA.DBF
CASDATA.FPT
CASDESC.DBF
CASEBK.DBF
CASEXEC.DBF
CASLEV1.DBF
CASLEV2.DBF
CASLEV3.DBF
CASLEV4.DBF
CASMBK.DBF
CASMISS.DBF

CASOCCS.DBF
CASOUTC.DBF
CASOUTC.FPT
CASPLAN.DBF
CASPREP.DBF
CASRECNO.DBF
CASREM.DBF
CASROTA.DBF
CASSCALE.DBF
CASTBK.DBF
CASTKDE.DBF
CASTRNG.DBF
CASUNIT.DBF
SEL_BOOK.DBF

5. Report forms:

CASOCMT.FRX
CASODAY.FRX
CASOMIS.FRX
CASOROT.FRX
CASRPT1.FRX
CASRPT2.FRX
CASRPT3.FRX

APPENDIX E

CLOSE AIR SUPPORT DATABASE MENU FILE SUMMARY

1. System: Close Air Support

2. Author: Dave Butterfield/Jerry Fargo

3. The Menu File Summary (CASMENU.MNX) lists the functions, commands, and procedures used in the CAS database menus.

CASMENU.MNX Last updated: 11/18/94 at 12:25:22

File		ALT+F	_msm_file
	New...		_mfi_new
	Open...		_mfi_open
	Close		_mfi_close
	Close All		_mfi_clall
	------------		_mfi_sp100
	Save		_mfi_save
	Save As...		_mfi_savas
	Revert		_mfi_revrt
	------------		_mfi_sp200
	Print Setup...		_mfi_setup
	Print...		_mfi_print
	------------		_mfi_sp300
	Exit		_mfi_quit
Edit		ALT+E	_msm_edit
	Undo	CTRL+Z	_med_undo
	Redo	CTRL+R	_med_redo
	------------		_med_sp100
	Cut	CTRL+X	_med_cut
	Copy	CTRL+C	_med_copy
	Paste	CTRL+V	_med_paste
	Paste Special...		_med_pstlk
	Clear		_med_clear
	------------		_med_sp300

	Select All	CTRL+A	_med_slcta
	------------		_med_sp400
	Goto Line...		_med_goto
	Find...	CTRL+F	_med_find
	Find Again	CTRL+G	_med_finda
	Replace And Find Again	CTRL+E	_med_repl
	Replace All		_med_repla
Database		ALT+D	_msm_data
	Setup...		_mda_setup
	Browse		_mda_brow
	------------		_mda_sp100
	Append From...		_mda_appnd
	Copy To...		_mda_copy
	Sort...		_mda_sort
	Total...		_mda_total
	------------		_mda_sp200
	Average...		_mda_avg
	Count...		_mda_count
	Sum...		_mda_sum
	Calculate...		_mda_calc
	Report...		_mda_reprt
	Label...		_mda_label
	------------		_mda_sp300
	Pack		_mda_pack
	Reindex		_mda_rindx
Record		ALT+R	_msm_recrd
	Append		_mrc_appnd
	Change		_mrc_chnge
	------------		_mrc_sp100
	Goto...		_mrc_goto
	Locate...		_mrc_locat
	Continue	CTRL+K	_mrc_cont
	Seek...		_mrc_seek
	------------		_mrc_sp200
	Replace...		_mrc_repl
	Delete...		_mrc_delet
	Recall...		_mrc_recal
Program		ALT+P	_msm_prog
	Do...	CTRL+D	_mpr_do
	Cancel		_mpr_cancl
	Resume	CTRL+M	_mpr_resum

Window	ALT+W	_msm_windo
Hide		_mwi_hide
Hide All		_mwi_hidea
Show All		_mwi_showa
Clear		_mwi_clear
Cycle	CTRL+F1	_mwi_rotat
------------		_mwi_sp100
Command	CTRL+F2	_mwi_cmd
View		_mwi_view
Close Air Support		CloseAirSu
Data Entry - CAS Master Database		DataEntryC
AFAC Planning		(Procedure)
AFAC Preparation		(Procedure)
MANEUVER Planning		(Procedure)
MANEUVER Preparation		(Procedure)
MANEUVER Execution		(Procedure)
TACP Planning		(Procedure)
TACP Preparation		(Procedure)
CAS Execution		(Procedure)
Data Entry - CAS Outcome Database		(Procedure)
Reports - CAS Master Database		(Procedure)
Reports - CAS Outcome Database		(Procedure)
ECI -> CAS Conversion		ECICASConv
CAS Master Database		(Procedure)
CAS Outcome Database		(Procedure)
Help	ALT+H	_msm_systm
Contents	F1	_mst_help
Search for Help on...		_mst_hpsch
How to Use Help		_mst_hphow
Calculator		_mst_calcu
Filer		_mst_filer

APPENDIX F

CLOSE AIR SUPPORT DATABASE PROCEDURE AND FUNCTION SUMMARY

1. System: Close Air Support

2. Author: Dave Butterfield/Jerry Fargo

3. The Procedure and Function Summary describes the procedures and functions used by different programs in the CAS database. It is organized by program and depicts each of the procedures, how it is called by the program, and what other programs or procedures are called by the procedure.

4. There are 7 program files in the system containing procedures. They are:

CASMENU.MPR	Menu definition and procedure file (Menu generated by GENMENU).
CASENTRY.PRG	Program to enter data into CAS database.
CASRPTS.PRG	Program to allow user interface to SQL query and reports.
CASCONV.PRG	Program to convert ECI data to CAS database format.
CASOUT1.PRG	Program to enter data into the CAS outcome database.
CASOROT.PRG	Program providing SQL queries for outcome reports.
CASORPTS.PRG	Program to allow user interface to SQL query for outcome reports.

----

1. CASMENU.MPR -- Last updated: 11/18/94 at 12:24:24

```
Contains: _QU70J2FB9            (Params: none)
          Called by: CASMENU.MPR
          Calls: CASOUT1.PRG
Contains: _QU70J2FCT            (Params: none)
          Called by: CASMENU.MPR
          Calls: CASRPTS.PRG
Contains: _QU70J2FEC            (Params: none)
          Called by: CASMENU.MPR
          Calls: CASORPTS.PRG
Contains: _QU70J2FLO            (Params: none)
          Called by: CASMENU.MPR
```

```
             Calls: CASENTRY.PRG
   Contains: _QU70J2FNM              (Params: none)
             Called by: CASMENU.MPR
             Calls: CASENTRY.PRG
   Contains: _QU70J2FPI              (Params: none)
             Called by: CASMENU.MPR
             Calls: CASENTRY.PRG
   Contains: _QU70J2FR2              (Params: none)
             Called by: CASMENU.MPR
             Calls: CASENTRY.PRG
   Contains: _QU70J2FSN              (Params: none)
             Called by: CASMENU.MPR
             Calls: CASENTRY.PRG
   Contains: _QU70J2FUJ              (Params: none)
             Called by: CASMENU.MPR
             Calls: CASENTRY.PRG
   Contains: _QU70J2FWF              (Params: none)
             Called by: CASMENU.MPR
             Calls: CASENTRY.PRG
   Contains: _QU70J2FYA              (Params: none)
             Called by: CASMENU.MPR
             Calls: CASENTRY.PRG
   Contains: _QU70J2G35              (Params: none)
             Called by: CASMENU.MPR
             Calls: CASCONV.PRG
   Contains: _QU70J2G4O              (Params: none)
             Called by: CASMENU.MPR
             Calls: CASCONV.PRG


--------------------------------------------------------------------------

2.         CASENTRY.PRG -- Last updated:  11/18/94 at 12:24:28

   Contains: _WIN_LOWER()            (Params: none)
             Called by: CASENTRY.PRG
             Called by: CASOUT1.PRG
             Called by: CASORPTS.PRG


--------------------------------------------------------------------------
```

```
3.      CASRPTS.PRG -- Last updated:  11/18/94 at 12:24:34

        Contains: _CREATE_RPT()          (Params: none)
                  Called by: CASRPTS.PRG
                  Calls: _GET_RPT        (procedure in CASRPTS.PRG)
                  Calls: _GET_BOOK       (procedure in CASRPTS.PRG)
                  Calls: _GET_TYPE       (procedure in CASRPTS.PRG)
                  Calls: _GET_ECHLEV     (procedure in CASRPTS.PRG)
                  Calls: CASCRPT1.PRG
                  Calls: CASNONE.PRG
                  Calls: CASLOC.PRG
                  Calls: CASCRPT2.PRG
                  Calls: CASCRPT3.PRG
        Contains: _CHKFORUPDATE          (Params: UPDATE_DB)
                  Called by: CASRPTS.PRG
        Contains: _WAITMSG               (Params: SHOWMSG)
                  Called by: CASRPTS.PRG
        Contains: _ROTATION              (Params: ROTA_ARRAY)
                  Called by: CASRPTS.PRG
        Contains: _MISSION               (Params: MISS_ARRAY)
                  Called by: CASRPTS.PRG
        Contains: _TRAINING              (Params: TRNG_ARRAY)
                  Called by: CASRPTS.PRG
        Contains: _CALLSIGN              (Params: OCCS_ARRAY)
                  Called by: CASRPTS.PRG
        Contains: _UNITOBS               (Params: UNIT_ARRAY)
                  Called by: CASRPTS.PRG
        Contains: _GET_RPT               (Params: SEL_RPT, SEL_TITLE1,
                                         SEL_TITLE2)
                  Called by: _CREATE_RPT() (function in CASRPTS.PRG)
        Contains: _GET_BOOK              (Params: SEL_BOOK, SEL_TITLE3,
                                         SEL_TKID)
                  Called by: _CREATE_RPT() (function in CASRPTS.PRG)
        Contains: _GET_TYPE              (Params: SEL_TYPE, SEL_TITLE4)
                  Called by: _CREATE_RPT() (function in CASRPTS.PRG)
        Contains: _GET_ECHLEV            (Params: SEL_ECHLEV, SEL_TITLE5)
                  Called by: _CREATE_RPT() (function in CASRPTS.PRG)
        Contains: _STOP_LOOP()           (Params: none)
                  Called by: CASRPTS.PRG
```

```
---------------------------------------------------------------------------

4.      CASCONV.PRG -- Last updated:  11/18/94 at 12:25:00

        Contains: _GET_TASKID         (Params: CURRENT_FIELD, MTASK_ID,
                                      MOD_TK_NO, MOD_REM)
                  Called by: CASCONV.PRG
        Contains: _CHK_TASKNO         (Params: CURRENT_FIELD,
                                      DATA_ARRAY, ROW_PTR, COL_PTR,
                                      COL_COUNT, MREMARKS,
                                      MTASK_NO, MSCORE, MOD_REM)
                  Called by: CASCONV.PRG
        Contains: _PUT_FIELDS         (Params: MROTATION, MTRNG_DAY,
                                      MTIME, MUNIT_OBS, MECHELON,
                                      MOC_CS, MMISSION, MCAS_MIS,
                                      MTASK_ID, MTASK_NO, MSCORE,
                                      MREMARKS)
                  Called by: CASCONV.PRG
        Contains: _WAITWINDOW         (Params: CURRENT_DBASE,
                                      IMPORT_DBASE)
                  Called by: CASCONV.PRG


---------------------------------------------------------------------------

5.      CASOUT1.PRG -- Last updated:  11/18/94 at 12:25:06

        Contains: _WIN_LOWER()        (Params: none)
                  Called by: CASENTRY.PRG
                  Called by: CASOUT1.PRG
                  Called by: CASORPTS.PRG
        Contains: _INVALID_DATA       (Params: none)


---------------------------------------------------------------------------
```

```
--------------------------------------------------------------------------

6.        CASOROT.PRG -- Last updated:  11/22/94 at  9:51:52

          Contains: _ROT_SUMMARY              (Params: none)
                    Called by: CASOROT.PRG
                    Calls: CASLOC.PRG
          Contains: _MIS_SUMMARY              (Params: none)
                    Called by: CASOROT.PRG
                    Calls: CASLOC.PRG
          Contains: _DAY_SUMMARY                  (Params: none)
                    Called by: CASOROT.PRG
                    Calls: CASLOC.PRG
          Contains: _CMT_SUMMARY              (Params: none)
                    Called by: CASOROT.PRG
                    Calls: CASLOC.PRG


--------------------------------------------------------------------------

7.        CASORPTS.PRG -- Last updated:  11/18/94 at 12:25:12

          Contains: _MAKE_ARRAYS              (Params: none)
                    Called by: CASORPTS.PRG
          Contains: _DOREPORTS()              (Params: none)
                    Called by: CASORPTS.PRG
                    Calls: CASOROT.PRG
          Contains: _WIN_LOWER()              (Params: none)
                    Called by: CASENTRY.PRG
                    Called by: CASOUT1.PRG
                    Called by: CASORPTS.PRG


--------------------------------------------------------------------------
```

APPENDIX G

CLOSE AIR SUPPORT DATABASE STRUCTURE SUMMARY

1. System: Close Air Support

2. Author: Dave Butterfield/Jerry Fargo

3. The Database/Table Structure Summary provides a summary of the fields,to include field size and position within the database, for each of the databases in the CAS database system. The summary also indicates which procedures use each database.

4. The 24 tables/databases in the system are listed below.

CAS AFAC Book Task Numbers	CASABK.DBF
CAS Master Database	CASDATA.DBF
CAS Task Number Description	CASDESC.DBF
CAS Execution Book Task Numbers	CASEBK.DBF
CAS Execution Task Numbers	CASEXEC.DBF
CAS Level One Task Numbers	CASLEV1.DBF
CAS Level Two Task Numbers	CASLEV2.DBF
CAS Level Three Task Numbers	CASLEV3.DBF
CAS Level Four Task Numbers	CASLEV4.DBF
CAS Maneuver Book Task Numbers	CASMBK.DBF
CAS Planning Task Numbers	CASPLAN.DBF
CAS Preparation Task Numbers	CASPREP.DBF
CAS Score Scale Descriptions	CASSCALE.DBF
CAS TACP Book Task Numbers	CASTBK.DBF
CAS Task ID Descriptions	CASTKDE.DBF
CAS Remark Task Numbers and Descriptions	CASREM.DBF
CAS Outcome Database	CASOUTC.DBF
CAS Mission Mission Description	CASMISS.DBF
CAS Observer/Controller Call Sign	CASOCCS.DBF
CAS Rotation Designation	CASROTA.DBF
CAS Training Day Description	CASTRNG.DBF
CAS Unit Designation	CASUNIT.DBF
CAS Number of records	CASRECNO.DBF
CAS Task Book Titles	SEL_BOOK.DBF

----

1. Structure for table/dbf: CASABK.DBF

Number of data records : 198
Last updated : 10/07/94

Field	Field name	Type	Width	Dec	Start	End
1	TASK_NO	Char	7	0	1	7
2	TK_ID_TYPE	Char	2	0	8	9
** Total **			10			

Used by:	_QU70J2FLO	(procedure in CASMENU.MPR)
	_QU70J2FNM	(procedure in CASMENU.MPR)

----

2. Structure for table/dbf: CASDATA.DBF

Number of data records : 0
Last updated : 10/11/94

Field	Field name	Type	Width	Dec	Start	End
1	ROTATION	Char	4	0	1	4
2	TRNG_DAY	Char	8	0	5	12
3	TIME	Char	4	0	13	16
4	ECHELON	Char	5	0	17	21
5	UNIT_OBS	Char	15	0	22	36
6	OC_CS	Char	7	0	37	43
7	MISSION	Char	15	0	44	58
8	CAS_MIS	Char	6	0	59	64
9	TASK_ID	Char	2	0	65	66
10	TASK_NO	Char	7	0	67	73
11	SCORE	Numeric	1	0	74	74
12	REMARKS	Memo	10	0	75	84
** Total **			85			

This table/dbf is associated with the memo file: CASDATA.FPT

Used by:	CASCRPT2.PRG	
	CASCRPT3.PRG	
	CASCRPT1.PRG	
	_ROTATION	(procedure in CASRPTS.PRG)
	_MISSION	(procedure in CASRPTS.PRG)
	_TRAINING	(procedure in CASRPTS.PRG)
	_CALLSIGN	(procedure in CASRPTS.PRG)
	_UNITOBS	(procedure in CASRPTS.PRG)

---

3. Structure for table/dbf: CASDESC.DBF

Number of data records : 980
Last updated : 10/10/94

Field	Field name	Type	Width	Dec	Start	End
1	TASK_NO	Char	7	0	1	7
2	TK_ID_TYPE	Char	2	0	8	9
3	TASK_DESC	Char	254	0	10	263
** Total **			264			

Used by: CASCRPT1.PRG

---

4. Structure for table/dbf: CASEBK.DBF

Number of data records : 136
Last updated : 10/07/94

Field	Field name	Type	Width	Dec	Start	End
1	TASK_NO	Char	7	0	1	7
2	TK_ID_TYPE	Char	2	0	8	9
** Total **			10			

Used by: _QU70J2FYA (procedure in CASMENU.MPR)

---

5. Structure for table/dbf: CASEXEC.DBF

Number of data records : 206
Last updated : 10/07/94

Field	Field name	Type	Width	Dec	Start	End
1	TASK_NO	Char	7	0	1	7
2	TK_ID_TYPE	Char	2	0	8	9
** Total **			10			

```
6.      Structure for table/dbf:    CASLEV1.DBF

        Number of data records :    121
        Last updated :              10/07/94
        Field     Field name        Type     Width    Dec     Start    End
        1         TASK_NO           Char     7        0       1        7
        ** Total **                          8

------------------------------------------------------------------------

7.      Structure for table/dbf:    CASLEV2.DBF

        Number of data records :    632
        Last updated :              10/10/94
        Field     Field name        Type     Width    Dec     Start    End
        1         TASK_NO           Char     7        0       1        7
        ** Total **                          8

------------------------------------------------------------------------

8.      Structure for table/dbf:    CASLEV3.DBF

        Number of data records :    851
        Last updated :              10/10/94
        Field     Field name        Type     Width    Dec     Start    End
        1         TASK_NO           Char     7        0       1        7
        ** Total **                          8

------------------------------------------------------------------------

9.      Structure for table/dbf:    CASLEV4.DBF

        Number of data records :    858
        Last updated :              10/10/94
        Field     Field name        Type     Width    Dec     Start    End
        1         TASK_NO           Char     7        0       1        7
        ** Total **                 8
```

---

10. Structure for table/dbf: CASMBK.DBF

Number of data records : 320
Last updated : 10/10/94

Field	Field name	Type	Width	Dec	Start	End
1	TASK_NO	Char	7	0	1	7
2	TK_ID_TYPE	Char	2	0	8	9
** Total **			10			

Used by:		
	_QU70J2FPI	(procedure in CASMENU.MPR)
	_QU70J2FR2	(procedure in CASMENU.MPR)
	_QU70J2FSN	(procedure in CASMENU.MPR)

---

11. Structure for table/dbf: CASPLAN.DBF

Number of data records : 564
Last updated : 10/07/94

Field	Field name	Type	Width	Dec	Start	End
1	TASK_NO	Char	7	0	1	7
2	TK_ID_TYPE	Char	2	0	8	9
** Total **			10			

---

12. Structure for table/dbf: CASPREP.DBF

Number of data records : 209
Last updated : 10/07/94

Field	Field name	Type	Width	Dec	Start	End
1	TASK_NO	Char	7	0	1	7
2	TK_ID_TYPE	Char	2	0	8	9
** Total **			10			

---

13. Structure for table/dbf: CASSCALE.DBF

Number of data records :		9				
Last updated :		07/26/94				
Field	Field name	Type	Width	Dec	Start	End
1	SCORE	Numeric	2	0	1	2
2	SHORT_DESC	Char	10	0	3	12
3	LONG_DESC	Char	20	0	13	32
** Total **			33			

Used by: CASCRPT1.PRG

----

14. Structure for table/dbf: CASTBK.DBF

Number of data records :		325				
Last updated :		10/07/94				
Field	Field name	Type	Width	Dec	Start	End
1	TASK_NO	Char	7	0	1	7
2	TK_ID_TYPE	Char	2	0	8	9
** Total **			10			

Last updated : 10/07/94

Used by:	_QU70J2FUJ	(procedure in CASMENU.MPR)
	_QU70J2FWF	(procedure in CASMENU.MPR)

----

15. Structure for table/dbf: CASTKDE.DBF

Number of data records :		9				
Last updated :		10/06/94				
Field	Field name	Type	Width	Dec	Start	End
1	TASK_ID	Char	2	0	1	2
2	TK_ID_DESC	Char	15	0	3	17
** Total **			18			

----

16. Structure for table/dbf: CASREM.DBF

Number of data records : 121
Last updated : 10/12/94

Field	Field name	Type	Width	Dec	Start	End
1	TASK_NO	Char	7	0	1	7
2	TK_ID_TYPE	Char	2	0	8	9
3	TASK_DESC	Char	254	0	10	263
** Total **			264			

Used by: CASCRPT3.PRG

---

17. Structure for table/dbf: CASOUTC.DBF

Number of data records : 0
Last updated : 10/07/94

Field	Field name	Type	Width	Dec	Start	End
1	ROTATION	Char	4	0	1	4
2	MISSION	Char	15	0	5	19
3	OC_CS	Char	8	0	20	27
4	DTG	Char	15	0	28	42
5	TRN_DAY	Char	4	0	43	46
6	LETH_A	Numeric	2	0	47	48
7	LETH_B	Numeric	2	0	49	50
8	SURV_A	Numeric	2	0	51	52
9	SURV_B	Numeric	2	0	53	54
10	COM_MIS	Numeric	1	0	55	55
11	COM_ENE	Numeric	1	0	56	56
12	COM_TRO	Numeric	1	0	57	57
13	COM_TER	Numeric	1	0	58	58
14	COM_TIM	Numeric	1	0	59	59
15	REM_MIS	Memo	10	0	60	69
16	REM_ENE	Memo	10	0	70	79
17	REM_TRO	Memo	10	0	80	89
18	REM_TER	Memo	10	0	90	99
19	REM_TIM	Memo	10	0	100	109
** Total **			110			

This table/dbf is associated with the memo file: CASOUTC.FPT

Used by: CASOROT.PRG
_MAKE_ARRAYS (procedure in CASORPTS.PRG)

```
------------------------------------------------------------------------

18.     Structure for table/dbf:     CASMISS.DBF

        Number of data records :     0
         Last updated :              10/11/94
        Field    Field name          Type    Width    Dec    Start    End
        1        MISSION             Char    15       0      1        15
        ** Total **                          16

        Used by: _MISSION            (procedure in CASRPTS.PRG)

------------------------------------------------------------------------

19.     Structure for table/dbf:     CASOCCS.DBF

        Number of data records :     0
        Last updated :               10/11/94
        Field    Field name          Type    Width    Dec    Start    End
        1        OC_CS               Char    7        0      1        7
        ** Total **                          8

        Used by: _CALLSIGN           (procedure in CASRPTS.PRG)

------------------------------------------------------------------------

20.     Structure for table/dbf:     CASROTA.DBF

        Number of data records :     0
        Last updated :               10/11/94
        Field    Field name          Type    Width    Dec    Start    End
        1        ROTATION            Char    4        0      1        4
        ** Total **                          5

        Used by: _ROTATION           (procedure in CASRPTS.PRG)

------------------------------------------------------------------------
```

21. Structure for table/dbf: CASTRNG.DBF

Number of data records :		0				
Last updated :		10/11/94				
Field	Field name	Type	Width	Dec	Start	End
1	TRNG_DAY	Char	8	0	1	8
** Total **			9			

Used by: _TRAINING (procedure in CASRPTS.PRG)

----

22. Structure for table/dbf: CASUNIT.DBF

Number of data records :		0				
Last updated :		10/11/94				
Field	Field name	Type	Width	Dec	Start	End
1	UNIT_OBS	Char	15	0	1	15
** Total **			16			

Used by: _UNITOBS (procedure in CASRPTS.PRG)

----

23. Structure for table/dbf: CASRECNO.DBF

Number of data records :		1				
Last updated :		10/11/94				
Field	Field name	Type	Width	Dec	Start	End
1	NUM_RECS	Numeric	10	0	1	10
** Total **			11			

----

24. Table/Database Field Concordance

Field Name	Type	Len	Dec	Table/DBF
CAS_MIS	C	6	0	CASDATA.DBF
COM_ENE	N	1	0	CASOUTC.DBF
COM_MIS	N	1	0	CASOUTC.DBF
COM_TER	N	1	0	CASOUTC.DBF
COM_TIM	N	1	0	CASOUTC.DBF
COM_TRO	N	1	0	CASOUTC.DBF
DTG	C	15	0	CASOUTC.DBF
ECHELON	C	5	0	CASDATA.DBF
LETH_A	N	2	0	CASOUTC.DBF
LETH_B	N	2	0	CASOUTC.DBF
LONG_DESC	C	20	0	CASSCALE.DBF
MISSION	C	15	0	CASDATA.DBF
MISSION	C	15	0	CASMISS.DBF
MISSION	C	15	0	CASOUTC.DBF
NUM_RECS	N	10	0	CASRECNO.DBF
OC_CS	C	7	0	CASDATA.DBF
OC_CS	C	7	0	CASOCCS.DBF
OC_CS	C	8	0	CASOUTC.DBF
REM_ENE	M	10	0	CASOUTC.DBF
REM_MIS	M	10	0	CASOUTC.DBF
REM_TER	M	10	0	CASOUTC.DBF
REM_TIM	M	10	0	CASOUTC.DBF
REM_TRO	M	10	0	CASOUTC.DBF
REMARKS	M	10	0	CASDATA.DBF
ROTATION	C	4	0	CASDATA.DBF
ROTATION	C	4	0	CASOUTC.DBF
ROTATION	C	4	0	CASROTA.DBF
SCORE	N	1	0	CASDATA.DBF
SCORE	N	2	0	CASSCALE.DBF
SHORT_DESC	C	10	0	CASSCALE.DBF
SURV_A	N	2	0	CASOUTC.DBF
SURV_B	N	2	0	CASOUTC.DBF
TASK_DESC	C	254	0	CASDESC.DBF
TASK_DESC	C	254	0	CASREM.DBF
TASK_ID	C	2	0	CASDATA.DBF
TASK_ID	C	2	0	CASTKDE.DBF
TASK_NO	C	7	0	CASABK.DBF
TASK_NO	C	7	0	CASDATA.DBF
TASK_NO	C	7	0	CASDESC.DBF

TASK_NO	C	7	0	CASEBK.DBF
TASK_NO	C	7	0	CASEXEC.DBF
TASK_NO	C	7	0	CASLEV1.DBF
TASK_NO	C	7	0	CASLEV2.DBF
TASK_NO	C	7	0	CASLEV3.DBF
TASK_NO	C	7	0	CASLEV4.DBF
TASK_NO	C	7	0	CASMBK.DBF
TASK_NO	C	7	0	CASPLAN.DBF
TASK_NO	C	7	0	CASPREP.DBF
TASK_NO	C	7	0	CASREM.DBF
TASK_NO	C	7	0	CASTBK.DBF
TIME	C	4	0	CASDATA.DBF
TK_ID_DESC	C	15	0	CASTKDE.DBF
TK_ID_TYPE	C	2	0	CASABK.DBF
TK_ID_TYPE	C	2	0	CASDESC.DBF
TK_ID_TYPE	C	2	0	CASEBK.DBF
TK_ID_TYPE	C	2	0	CASEXEC.DBF
TK_ID_TYPE	C	2	0	CASMBK.DBF
TK_ID_TYPE	C	2	0	CASPLAN.DBF
TK_ID_TYPE	C	2	0	CASPREP.DBF
TK_ID_TYPE	C	2	0	CASREM.DBF
TK_ID_TYPE	C	2	0	CASTBK.DBF
TRN_DAY	C	4	0	CASOUTC.DBF
TRNG_DAY	C	8	0	CASDATA.DBF
TRNG_DAY	C	8	0	CASTRNG.DBF
UNIT_OBS	C	15	0	CASDATA.DBF
UNIT_OBS	C	15	0	CASUNIT.DBF

APPENDIX H

CLOSE AIR SUPPORT REPORT FORM SUMMARY

1. System: Close Air Support

2. Author: Dave Butterfield/Jerry Fargo

3. The Report Form File Summary identifies the 7 report forms in the system. It also identifies the procedure or function that uses the report form and the program which contains that procedure. The report forms are listed below.

   CASRPT2.FRX
   CASRPT3.FRX
   CASOCMT.FRX
   CASODAY.FRX
   CASOMIS.FRX
   CASOROT.FRX
   CASRPT1.FRX

----

1. CASRPT2.FRX

Last updated:	05/11/94 at 14:49:20	
Used by:	_CREATE_RPT()	(function in CASRPTS.PRG)

----

2. CASRPT3.FRX

Last updated:	08/18/94 at 12:23:24	
Used by:	_CREATE_RPT()	(function in CASRPTS.PRG)

----

3. CASOCMT.FRX

Last updated:	08/19/94 at 13:22:30	
Used by:	_CMT_SUMMARY	(procedure in CASOROT.PRG)

----

4. CASODAY.FRX

Last updated:	08/19/94 at 12:20:08	
Used by:	_DAY_SUMMARY	(procedure in CASOROT.PRG)

----

5. CASOMIS.FRX

Last updated:	08/18/94 at 15:45:44	
Used by:	_MIS_SUMMARY	(procedure in CASOROT.PRG)

----

6. CASOROT.FRX

Last updated:	08/19/94 at 9:22:36	
Used by:	_ROT_SUMMARY	(procedure in CASOROT.PRG)

----

7. CASRPT1.FRX

Last updated:	05/03/94 at 12:40:38	
Used by:	_CREATE_RPT()	(function in CASRPTS.PRG)

APPENDIX I

CLOSE AIR SUPPORT MENU DEFINITION AND PROCEDURE FILE

1. Procedure file: C:\TEMP\CASMENU.MPR

2. System: Close Air Support

3. Author: Dave Butterfield/Jerry Fargo

4. Last modified: 11/18/94 at 12:24:24

5. Procedures & Functions:

   _QU70J2FB9
   _QU70J2FCT
   _QU70J2FEC
   _QU70J2FLO
   _QU70J2FNM
   _QU70J2FPI
   _QU70J2FR2
   _QU70J2FSN
   _QU70J2FUJ
   _QU70J2FWF
   _QU70J2FYA
   _QU70J2G35
   _QU70J2G4O

6. Calls:

_QU70J2FB9	(procedure in CASMENU.MPR)
_QU70J2FCT	(procedure in CASMENU.MPR)
_QU70J2FEC	(procedure in CASMENU.MPR)
_QU70J2FLO	(procedure in CASMENU.MPR)
_QU70J2FNM	(procedure in CASMENU.MPR)
_QU70J2FPI	(procedure in CASMENU.MPR)
_QU70J2FR2	(procedure in CASMENU.MPR)
_QU70J2FSN	(procedure in CASMENU.MPR)
_QU70J2FUJ	(procedure in CASMENU.MPR)
_QU70J2FWF	(procedure in CASMENU.MPR)
_QU70J2FYA	(procedure in CASMENU.MPR)

```
                    _QU70J2G35      (procedure in CASMENU.MPR)
                    _QU70J2G4O      (procedure in CASMENU.MPR)


*    ******************************************************
*    *
*    * Description:
*    * This program was automatically generated by GENMENU.
*    *
*    ******************************************************


*    ******************************************************
*    *
*    *                Menu Definition
*    *
*    ******************************************************
*

SET SYSMENU TO

SET SYSMENU AUTOMATIC

DEFINE PAD _msm_file OF _msysmenu PROMPT "\<File" COLOR SCHEME 3 ;
  KEY alt+F, "" ;
  MESSAGE "Create, open, save, print files or quit FoxPro"
DEFINE PAD _msm_edit OF _msysmenu PROMPT "\<Edit" COLOR SCHEME 3 ;
  KEY alt+E, "" ;
  MESSAGE "Edit text or manipulate OLE objects"
DEFINE PAD _msm_data OF _msysmenu PROMPT "\<Database" COLOR SCHEME 3 ;
  KEY alt+D, "" ;
  MESSAGE "Perform operations on tables, print reports and labels"
DEFINE PAD _msm_recrd OF _msysmenu PROMPT "\<Record" COLOR SCHEME 3 ;
  KEY alt+R, "" ;
  MESSAGE "Perform operations on records in active table"
DEFINE PAD _msm_prog OF _msysmenu PROMPT "\<Program" COLOR SCHEME 3 ;
  KEY alt+p, "" ;
  MESSAGE "Debug, run, compile, generate and document programs"
DEFINE PAD _msm_windo OF _msysmenu PROMPT "\<Window" COLOR SCHEME 3 ;
  KEY alt+W, "" ;
  MESSAGE "Manipulate windows, display Command and View windows"
DEFINE PAD _qu70j2drp OF _msysmenu PROMPT "\<Close Air Support" COLOR
SCHEME 3 ;
  MESSAGE "Close Air Support Program."
DEFINE PAD _msm_systm OF _msysmenu PROMPT "\<Help" COLOR SCHEME 3 ;
```

```
  KEY alt+H, "" ;
  MESSAGE "Access information for learning and using FoxPro"
ON PAD _msm_file OF _msysmenu ACTIVATE POPUP _mfile
ON PAD _msm_edit OF _msysmenu ACTIVATE POPUP _medit
ON PAD _msm_data OF _msysmenu ACTIVATE POPUP _mdata
ON PAD _msm_recrd OF _msysmenu ACTIVATE POPUP _mrecord
ON PAD _msm_prog OF _msysmenu ACTIVATE POPUP _mprog
ON PAD _msm_windo OF _msysmenu ACTIVATE POPUP _mwindow
ON PAD _qu70j2drp OF _msysmenu ACTIVATE POPUP closeairsu
ON PAD _msm_systm OF _msysmenu ACTIVATE POPUP _msystem

DEFINE POPUP _mfile MARGIN RELATIVE SHADOW COLOR SCHEME 4
DEFINE BAR _mfi_new OF _mfile PROMPT "\<New..." ;
  MESSAGE "Create a new file"
DEFINE BAR _mfi_open OF _mfile PROMPT "\<Open..." ;
  MESSAGE "Open an existing file"
DEFINE BAR _mfi_close OF _mfile PROMPT "\<Close" ;
  MESSAGE "Close the frontmost file"
DEFINE BAR _mfi_clall OF _mfile PROMPT "Close All" ;
  MESSAGE "Close all files"
DEFINE BAR _mfi_sp100 OF _mfile PROMPT "\-"
DEFINE BAR _mfi_save OF _mfile PROMPT "\<Save" ;
  MESSAGE "Save the current file"
DEFINE BAR _mfi_savas OF _mfile PROMPT "Sa\<ve As..." ;
  MESSAGE "Save the current file with a new name"
DEFINE BAR _mfi_revrt OF _mfile PROMPT "\<Revert" ;
  MESSAGE "Revert to last saved version of file"
DEFINE BAR _mfi_sp200 OF _mfile PROMPT "\-"
DEFINE BAR _mfi_setup OF _mfile PROMPT "Pr\<int Setup..." ;
  MESSAGE "Specify printer and print options"
DEFINE BAR _mfi_print OF _mfile PROMPT "\<Print..." ;
  MESSAGE "Print text file, contents of the Command window or clipboard"
DEFINE BAR _mfi_sp300 OF _mfile PROMPT "\-"
DEFINE BAR _mfi_quit OF _mfile PROMPT "E\<xit" ;
  MESSAGE "Exit FoxPro"

DEFINE POPUP _medit MARGIN RELATIVE SHADOW COLOR SCHEME 4
DEFINE BAR _med_undo OF _medit PROMPT "\<Undo" ;
  KEY ctrl+z, "Ctrl+Z" ;
  MESSAGE "Reverse the most recent edit action"
DEFINE BAR _med_redo OF _medit PROMPT "\<Redo" ;
  KEY ctrl+R, "Ctrl+R" ;
  MESSAGE "Repeat the action previously reversed with Undo"
DEFINE BAR _med_sp100 OF _medit PROMPT "\-"
DEFINE BAR _med_cut OF _medit PROMPT "Cu\<t" ;
```

```
   KEY ctrl+x, "Ctrl+X" ;
   MESSAGE "Remove selection and put it on the clipboard"
DEFINE BAR _med_copy OF _medit PROMPT "\<Copy" ;
   KEY ctrl+C, "Ctrl+C" ;
   MESSAGE "Copy selection and put it on the clipboard"
DEFINE BAR _med_paste OF _medit PROMPT "\<Paste" ;
   KEY ctrl+v, "Ctrl+V" ;
   MESSAGE "Paste contents of the clipboard at the insertion point"
DEFINE BAR _med_pstlk OF _medit PROMPT "Paste \<Special..." ;
   MESSAGE "Establish link to copied data"
DEFINE BAR _med_clear OF _medit PROMPT "Clear" ;
   MESSAGE "Erase selection"
DEFINE BAR _med_sp300 OF _medit PROMPT "\-"
DEFINE BAR _med_slcta OF _medit PROMPT "Select \<All" ;
   KEY ctrl+A, "Ctrl+A" ;
   MESSAGE "Select all lines of text or objects in current window"
DEFINE BAR _med_sp400 OF _medit PROMPT "\-"
DEFINE BAR _med_goto OF _medit PROMPT "Goto \<Line..." ;
   MESSAGE "Move cursor to designated line number"
DEFINE BAR _med_find OF _medit PROMPT "\<Find..." ;
   KEY ctrl+F, "Ctrl+F" ;
   MESSAGE "Search for text"
DEFINE BAR _med_finda OF _medit PROMPT "Find A\<gain" ;
   KEY ctrl+G, "Ctrl+G" ;
   MESSAGE "Repeat the last text search"
DEFINE BAR _med_repl OF _medit PROMPT "R\<eplace And Find Again" ;
   KEY ctrl+E, "Ctrl+E" ;
   MESSAGE "Replace text and continue search"
DEFINE BAR _med_repla OF _medit PROMPT "Replace All" ;
   MESSAGE "Replace all occurrences of the specified text"

DEFINE POPUP _mdata MARGIN RELATIVE SHADOW COLOR SCHEME 4
DEFINE BAR _mda_setup OF _mdata PROMPT "Set\<up..." ;
   MESSAGE "Establish settings for table in the current work area"
DEFINE BAR _mda_brow OF _mdata PROMPT "\<Browse" ;
   MESSAGE "Examine and/or edit active table"
DEFINE BAR _mda_sp100 OF _mdata PROMPT "\-"
DEFINE BAR _mda_appnd OF _mdata PROMPT "\<Append From..." ;
   MESSAGE "Add records from another table"
DEFINE BAR _mda_copy OF _mdata PROMPT "\<Copy To..." ;
   MESSAGE "Copy contents of a table to a new file"
DEFINE BAR _mda_sort OF _mdata PROMPT "\<Sort..." ;
   MESSAGE "Sort a table"
DEFINE BAR _mda_total OF _mdata PROMPT "\<Total..." ;
   MESSAGE "Compute totals for numeric fields"
```

```
DEFINE BAR _mda_sp200 OF _mdata PROMPT "\-"
DEFINE BAR _mda_avg OF _mdata PROMPT "A\<verage..." ;
  MESSAGE "Compute the average for numeric fields"
DEFINE BAR _mda_count OF _mdata PROMPT "C\<ount..." ;
  MESSAGE "Count the number of table records"
DEFINE BAR _mda_sum OF _mdata PROMPT "Su\<m..." ;
  MESSAGE "Calculate the sum of numeric fields"
DEFINE BAR _mda_calc OF _mdata PROMPT "Calculat\<e..." ;
  MESSAGE "Perform statistical operations"
DEFINE BAR _mda_reprt OF _mdata PROMPT "\<Report..." ;
  MESSAGE "Display and print reports"
DEFINE BAR _mda_label OF _mdata PROMPT "\<Label..." ;
  MESSAGE "Display and print labels"
DEFINE BAR _mda_sp300 OF _mdata PROMPT "\-"
DEFINE BAR _mda_pack OF _mdata PROMPT "\<Pack" ;
  MESSAGE "Permanently remove records marked for deletion"
DEFINE BAR _mda_rindx OF _mdata PROMPT "Reinde\<x" ;
  MESSAGE "Rebuild active index files"

DEFINE POPUP _mrecord MARGIN RELATIVE SHADOW COLOR SCHEME 4
DEFINE BAR _mrc_appnd OF _mrecord PROMPT "\<Append" ;
  MESSAGE "Add a new record"
DEFINE BAR _mrc_chnge OF _mrecord PROMPT "Chang\<e" ;
  MESSAGE "Edit table records"
DEFINE BAR _mrc_sp100 OF _mrecord PROMPT "\-"
DEFINE BAR _mrc_goto OF _mrecord PROMPT "\<Goto..." ;
  MESSAGE "Go to a specific record"
DEFINE BAR _mrc_locat OF _mrecord PROMPT "\<Locate..." ;
  MESSAGE "Look for the record that matches a specified condition"
DEFINE BAR _mrc_cont OF _mrecord PROMPT "\<Continue" ;
  KEY ctrl+K, "Ctrl+K" ;
  MESSAGE "Continue to locate records"
DEFINE BAR _mrc_seek OF _mrecord PROMPT "\<Seek..." ;
  MESSAGE "Search an indexed table"
DEFINE BAR _mrc_sp200 OF _mrecord PROMPT "\-"
DEFINE BAR _mrc_repl OF _mrecord PROMPT "Re\<place..." ;
  MESSAGE "Update field information in a table"
DEFINE BAR _mrc_delet OF _mrecord PROMPT "\<Delete..." ;
  MESSAGE "Mark records for deletion"
DEFINE BAR _mrc_recal OF _mrecord PROMPT "\<Recall..." ;
  MESSAGE "Unmark records that are marked for deletion"

DEFINE POPUP _mprog MARGIN RELATIVE SHADOW COLOR SCHEME 4
DEFINE BAR _mpr_do OF _mprog PROMPT "\<Do..." ;
  KEY ctrl+D, "Ctrl+D" ;
```

```
   MESSAGE "Run a program"
DEFINE BAR _mpr_cancl OF _mprog PROMPT "\<Cancel" ;
   MESSAGE "Stop running a program"
DEFINE BAR _mpr_resum OF _mprog PROMPT "\<Resume" ;
   KEY ctrl+m, "Ctrl+M" ;
   MESSAGE "Resume suspended program"

DEFINE POPUP _mwindow MARGIN RELATIVE SHADOW COLOR SCHEME 4
DEFINE BAR _mwi_hide OF _mwindow PROMPT "\<Hide" ;
   MESSAGE "Remove active window from sight"
DEFINE BAR _mwi_hidea OF _mwindow PROMPT "Hide All" ;
   MESSAGE "Remove all windows from sight"
DEFINE BAR _mwi_showa OF _mwindow PROMPT "Sh\<ow All" ;
   MESSAGE "Show all hidden windows"
DEFINE BAR _mwi_clear OF _mwindow PROMPT "Clea\<r" ;
   MESSAGE "Clear current output window"
DEFINE BAR _mwi_rotat OF _mwindow PROMPT "\<Cycle" ;
   KEY ctrl+f1, "Ctrl+F1" ;
   MESSAGE "Rearrange open windows to bring successive ones forward"
DEFINE BAR _mwi_sp100 OF _mwindow PROMPT "\-"
DEFINE BAR _mwi_cmd OF _mwindow PROMPT "Co\<mmand" ;
   KEY ctrl+f2, "Ctrl+F2" ;
   MESSAGE "Display Command window"
DEFINE BAR _mwi_view OF _mwindow PROMPT "\<View" ;
   MESSAGE "Display the View window"

DEFINE POPUP closeairsu MARGIN RELATIVE SHADOW COLOR SCHEME 4
DEFINE BAR 1 OF closeairsu PROMPT "Data Entry - CAS Master Database" ;
   MESSAGE "Enter CAS Data from the CAS Forms."
DEFINE BAR 2 OF closeairsu PROMPT "Data Entry - CAS Outcome Database" ;
   MESSAGE "Enter the Outcome Data from the Forms."
DEFINE BAR 3 OF closeairsu PROMPT "Reports - CAS Master Database" ;
   MESSAGE "Initiate the CAS Report Generator."
DEFINE BAR 4 OF closeairsu PROMPT "Reports - CAS Outcome Database" ;
   MESSAGE "Initiate the CAS Outcome Report Generator"
DEFINE BAR 5 OF closeairsu PROMPT "ECI -> CAS Conversion" ;
   MESSAGE "Convert ECI Delimited Text Files."
ON BAR 1 OF closeairsu ACTIVATE POPUP dataentryc
ON SELECTION BAR 2 OF closeairsu ;
   DO _qu70j2fb9 ;
   IN LOCFILE("CAS\CASMENU" ,"MPX;MPR|FXP;PRG" ,"Where is CASMENU?")
ON SELECTION BAR 3 OF closeairsu ;
   DO _qu70j2fct ;
   IN LOCFILE("CAS\CASMENU" ,"MPX;MPR|FXP;PRG" ,"Where is CASMENU?")
ON SELECTION BAR 4 OF closeairsu ;
```

```
  DO _qu70j2fec ;
  IN LOCFILE("CAS\CASMENU" ,"MPX;MPR|FXP;PRG" ,"Where is CASMENU?")
ON BAR 5 OF closeairsu ACTIVATE POPUP ecicasconv

DEFINE POPUP dataentryc MARGIN RELATIVE SHADOW COLOR SCHEME 4
DEFINE BAR 1 OF dataentryc PROMPT "AFAC Planning"
DEFINE BAR 2 OF dataentryc PROMPT "AFAC Preparation"
DEFINE BAR 3 OF dataentryc PROMPT "MANEUVER Planning"
DEFINE BAR 4 OF dataentryc PROMPT "MANEUVER Preparation"
DEFINE BAR 5 OF dataentryc PROMPT "MANEUVER Execution"
DEFINE BAR 6 OF dataentryc PROMPT "TACP Planning"
DEFINE BAR 7 OF dataentryc PROMPT "TACP Preparation"
DEFINE BAR 8 OF dataentryc PROMPT "CAS Execution"
ON SELECTION BAR 1 OF dataentryc ;
  DO _qu70j2flo ;
  IN LOCFILE("CAS\CASMENU" ,"MPX;MPR|FXP;PRG" ,"Where is CASMENU?")
ON SELECTION BAR 2 OF dataentryc ;
  DO _qu70j2fnm ;
  IN LOCFILE("CAS\CASMENU" ,"MPX;MPR|FXP;PRG" ,"Where is CASMENU?")
ON SELECTION BAR 3 OF dataentryc ;
  DO _qu70j2fpi ;
  IN LOCFILE("CAS\CASMENU" ,"MPX;MPR|FXP;PRG" ,"Where is CASMENU?")
ON SELECTION BAR 4 OF dataentryc ;
  DO _qu70j2fr2 ;
  IN LOCFILE("CAS\CASMENU" ,"MPX;MPR|FXP;PRG" ,"Where is CASMENU?")
ON SELECTION BAR 5 OF dataentryc ;
  DO _qu70j2fsn ;
  IN LOCFILE("CAS\CASMENU" ,"MPX;MPR|FXP;PRG" ,"Where is CASMENU?")
ON SELECTION BAR 6 OF dataentryc ;
  DO _qu70j2fuj ;
  IN LOCFILE("CAS\CASMENU" ,"MPX;MPR|FXP;PRG" ,"Where is CASMENU?")
ON SELECTION BAR 7 OF dataentryc ;
  DO _qu70j2fwf ;
  IN LOCFILE("CAS\CASMENU" ,"MPX;MPR|FXP;PRG" ,"Where is CASMENU?")
ON SELECTION BAR 8 OF dataentryc ;
  DO _qu70j2fya ;
  IN LOCFILE("CAS\CASMENU" ,"MPX;MPR|FXP;PRG" ,"Where is CASMENU?")

DEFINE POPUP ecicasconv MARGIN RELATIVE SHADOW COLOR SCHEME 4
DEFINE BAR 1 OF ecicasconv PROMPT "CAS Master Database"
DEFINE BAR 2 OF ecicasconv PROMPT "CAS Outcome Database"
ON SELECTION BAR 1 OF ecicasconv ;
  DO _qu70j2g35 ;
  IN LOCFILE("CAS\CASMENU" ,"MPX;MPR|FXP;PRG" ,"Where is CASMENU?")
ON SELECTION BAR 2 OF ecicasconv ;
```

```
  DO _qu70j2g4o ;
  IN LOCFILE("CAS\CASMENU" ,"MPX;MPR|FXP;PRG" ,"Where is CASMENU?")

DEFINE POPUP _msystem MARGIN RELATIVE SHADOW COLOR SCHEME 4
DEFINE BAR _mst_help OF _msystem PROMPT "\<Contents" ;
  KEY f1, "" ;
  MESSAGE "Display help contents"
DEFINE BAR _mst_hpsch OF _msystem PROMPT "\<Search for Help on..." ;
  MESSAGE "Search for help topic by typing or selecting a keyword"
DEFINE BAR _mst_hphow OF _msystem PROMPT "\<How to Use Help" ;
  MESSAGE "Display instructions for using help"
DEFINE BAR _mst_calcu OF _msystem PROMPT "Ca\<lculator" ;
  MESSAGE "Perform calculations"
DEFINE BAR _mst_filer OF _msystem PROMPT "\<Filer" ;
  MESSAGE "Manage files and directories"


*       ********************************************************
*       *
*       * _QU70J2FB9  ON SELECTION BAR 2 OF POPUP closeairsu
*       *
*       * Procedure Origin:
*       *
*       * From Menu:  CASMENU.MPR,          Record:   95
*       * Called By:  ON SELECTION BAR 2 OF POPUP closeairsu
*       * Prompt:     Data Entry - CAS Outcome Database
*       * Snippet:    1
*       *
*       ********************************************************
*
*!***********************************************************************
*
*!
*!      Procedure: _QU70J2FB9
*!
*!      Called by: CASMENU.MPR
*!
*!          Calls: CASOUT1.PRG
*!
*!***********************************************************************
*
PROCEDURE _qu70j2fb9
DO casout1
```

```
*       *****************************************************
*       *
*       * _QU70J2FCT  ON SELECTION BAR 3 OF POPUP closeairsu
*       *
*       * Procedure Origin:
*       *
*       * From Menu:  CASMENU.MPR,           Record:   96
*       * Called By:  ON SELECTION BAR 3 OF POPUP closeairsu
*       * Prompt:     Reports - CAS Master Database
*       * Snippet:    2
*       *
*       *****************************************************
*
*!*********************************************************************
*
*!
*!      Procedure: _QU70J2FCT
*!
*!      Called by: CASMENU.MPR
*!
*!         Calls: CASRPTS.PRG
*!
*!*********************************************************************
*
PROCEDURE _qu70j2fct
DO casrpts


*       *****************************************************
*       *
*       * _QU70J2FEC  ON SELECTION BAR 4 OF POPUP closeairsu
*       *
*       * Procedure Origin:
*       *
*       * From Menu:  CASMENU.MPR,           Record:   97
*       * Called By:  ON SELECTION BAR 4 OF POPUP closeairsu
*       * Prompt:     Reports - CAS Outcome Database
*       * Snippet:    3
*       *
*       *****************************************************
*
*!*********************************************************************
*
*!
*!      Procedure: _QU70J2FEC
```

```
*!
*!     Called by: CASMENU.MPR
*!
*!          Calls: CASORPTS.PRG
*!
*!************************************************************************
*
PROCEDURE _qu70j2fec
DO casorpts



*       ****************************************************
*       *
*       * _QU70J2FLO  ON SELECTION BAR 1 OF POPUP dataentryc
*       *
*       * Procedure Origin:
*       *
*       * From Menu:  CASMENU.MPR,            Record:   87
*       * Called By:  ON SELECTION BAR 1 OF POPUP dataentryc
*       * Prompt:     AFAC Planning
*       * Snippet:    4
*       *
*       ****************************************************
*
*!************************************************************************
*
*!
*!     Procedure: _QU70J2FLO
*!
*!     Called by: CASMENU.MPR
*!
*!          Calls: CASENTRY.PRG
*!
*!           Uses: CASABK.DBF
*!
*!************************************************************************
*
PROCEDURE _qu70j2flo
*
* initialize selection specific variables
*
STORE 'A' TO mtask_id
STORE 'AFAC' TO mtk_id_desc
STORE 'PL' TO mtk_id_type
```

```
*
* get the selected task number questions
*
SELECT a.task_no ;
   FROM casabk A ;
   INTO ARRAY tmparray ;
   WHERE a.tk_id_type = 'PL'
*
* save the number of elements for use
*
num_elements = ALEN(tmparray)
*
* do the data entry program
*
DO casentry



*       *****************************************************
*       *
*       * _QU70J2FNM  ON SELECTION BAR 2 OF POPUP dataentryc
*       *
*       * Procedure Origin:
*       *
*       * From Menu:  CASMENU.MPR,          Record:   88
*       * Called By:  ON SELECTION BAR 2 OF POPUP dataentryc
*       * Prompt:     AFAC Preparation
*       * Snippet:    5
*       *
*       *****************************************************
*
*!*********************************************************************
*
*!
*!      Procedure: _QU70J2FNM
*!
*!      Called by: CASMENU.MPR
*!
*!          Calls: CASENTRY.PRG
*!
*!           Uses: CASABK.DBF
*!
*!*********************************************************************
*
PROCEDURE _qu70j2fnm
```

```
*
* initialize selection specific variables
*
STORE 'A' TO mtask_id
STORE 'AFAC' TO mtk_id_desc
STORE 'PR' TO mtk_id_type
*
* get the selected task number questions
*
SELECT a.task_no ;
   FROM casabk A ;
   INTO ARRAY tmparray ;
   WHERE a.tk_id_type = 'PR'
*
* save the number of elements for use
*
num_elements = ALEN(tmparray)
*
* do the data entry program
*
DO casentry


*       ********************************************************
*       *
*       * _QU70J2FPI  ON SELECTION BAR 3 OF POPUP dataentryc
*       *
*       * Procedure Origin:
*       *
*       * From Menu:  CASMENU.MPR,            Record:   89
*       * Called By:  ON SELECTION BAR 3 OF POPUP dataentryc
*       * Prompt:     MANEUVER Planning
*       * Snippet:    6
*       *
*       ********************************************************
*
*!**********************************************************************
*
*!
*!      Procedure: _QU70J2FPI
*!
*!      Called by: CASMENU.MPR
*!
*!          Calls: CASENTRY.PRG
```

```
*!
*!          Uses: CASMBK.DBF
*!
*!**************************************************************************
*
PROCEDURE _qu70j2fpi
*
* initialize selection specific variables
*
STORE 'M' TO mtask_id
STORE 'MANEUVER' TO mtk_id_desc
STORE 'PL' TO mtk_id_type
*
* get the selected task number questions
*
SELECT a.task_no ;
   FROM casmbk A ;
   INTO ARRAY tmparray ;
   WHERE a.tk_id_type = mtk_id_type
*
* save the number of elements for use
*
num_elements = ALEN(tmparray)
*
* do the data entry program
*
DO casentry


*       ******************************************************
*       *
*       * _QU70J2FR2  ON SELECTION BAR 4 OF POPUP dataentryc
*       *
*       * Procedure Origin:
*       *
*       * From Menu:  CASMENU.MPR,           Record:   90
*       * Called By:  ON SELECTION BAR 4 OF POPUP dataentryc
*       * Prompt:     MANEUVER Preparation
*       * Snippet:    7
*       *
*       ******************************************************
*
*!**************************************************************************
*
*!
```

```
*!      Procedure: _QU70J2FR2
*!
*!      Called by: CASMENU.MPR
*!
*!          Calls: CASENTRY.PRG
*!
*!           Uses: CASMBK.DBF
*!
*!*************************************************************************
*
PROCEDURE _qu70j2fr2
STORE 'M' TO mtask_id
STORE 'MANEUVER' TO mtk_id_desc
STORE 'PR' TO mtk_id_type
*
* get the selected task number questions
*
SELECT a.task_no ;
  FROM casmbk A ;
  INTO ARRAY tmparray ;
  WHERE a.tk_id_type = mtk_id_type
*
* save the number of elements for use
*
num_elements = ALEN(tmparray)
*
* do the data entry program
*
DO casentry


*       ******************************************************
*       *
*       * _QU70J2FSN  ON SELECTION BAR 5 OF POPUP dataentryc
*       *
*       * Procedure Origin:
*       *
*       * From Menu:  CASMENU.MPR,           Record:   91
*       * Called By:  ON SELECTION BAR 5 OF POPUP dataentryc
*       * Prompt:     MANEUVER Execution
*       * Snippet:    8
*       *
*       ******************************************************
*
*!*************************************************************************
```

```
*
*!
*!      Procedure: _QU70J2FSN
*!
*!      Called by: CASMENU.MPR
*!
*!          Calls: CASENTRY.PRG
*!
*!           Uses: CASMBK.DBF
*!
*!**************************************************************************
*
PROCEDURE _qu70j2fsn
STORE 'M' TO mtask_id
STORE 'MANEUVER' TO mtk_id_desc
STORE 'EX' TO mtk_id_type
*
* get the selected task number questions
*
SELECT a.task_no ;
   FROM casmbk A ;
   INTO ARRAY tmparray ;
   WHERE a.tk_id_type = mtk_id_type
*
* save the number of elements for use
*
num_elements = ALEN(tmparray)
*
* do the data entry program
*
DO casentry


*       ****************************************************
*       *
*       * _QU70J2FUJ  ON SELECTION BAR 6 OF POPUP dataentryc
*       *
*       * Procedure Origin:
*       *
*       * From Menu:  CASMENU.MPR,          Record:   92
*       * Called By:  ON SELECTION BAR 6 OF POPUP dataentryc
*       * Prompt:     TACP Planning
*       * Snippet:    9
*       *
*       ****************************************************
```

```
*
*!*************************************************************************
*
*!
*!      Procedure: _QU70J2FUJ
*!
*!      Called by: CASMENU.MPR
*!
*!          Calls: CASENTRY.PRG
*!
*!           Uses: CASTBK.DBF
*!
*!*************************************************************************
*
PROCEDURE _qu70j2fuj
*
* initialize selection specific variables
*
STORE 'G' TO mtask_id
STORE 'TACP' TO mtk_id_desc
STORE 'PL' TO mtk_id_type
*
* get the selected task number questions
*
SELECT a.task_no ;
   FROM castbk A ;
   INTO ARRAY tmparray ;
   WHERE a.tk_id_type = 'PL'
*
* save the number of elements for use
*
num_elements = ALEN(tmparray)
*
* do the data entry program
*
DO casentry


*       ******************************************************
*       *
*       * _QU70J2FWF  ON SELECTION BAR 7 OF POPUP dataentryc
*       *
*       * Procedure Origin:
*       *
*       * From Menu:  CASMENU.MPR,          Record:   93
*       * Called By:  ON SELECTION BAR 7 OF POPUP dataentryc
*       * Prompt:     TACP Preparation
*       * Snippet:    10
*       *
*       ********************************************************
*
*!**********************************************************************
*
*!
*!      Procedure: _QU70J2FWF
*!
*!      Called by: CASMENU.MPR
*!
*!          Calls: CASENTRY.PRG
*!
*!           Uses: CASTBK.DBF
*!
*!**********************************************************************
*
PROCEDURE _qu70j2fwf
*
* initialize selection specific variables
*
STORE 'G' TO mtask_id
STORE 'TACP' TO mtk_id_desc
STORE 'PR' TO mtk_id_type
*
* get the selected task number questions
*
SELECT a.task_no ;
  FROM castbk A ;
  INTO ARRAY tmparray ;
  WHERE a.tk_id_type = 'PR'
*
* save the number of elements for use
*
num_elements = ALEN(tmparray)
*
* do the data entry program
*
DO casentry


*       ********************************************************
*       *
*       * _QU70J2FYA  ON SELECTION BAR 8 OF POPUP dataentryc
*       *
*       * Procedure Origin:
*       *
*       * From Menu:  CASMENU.MPR,           Record:   94
*       * Called By:  ON SELECTION BAR 8 OF POPUP dataentryc
*       * Prompt:     CAS Execution
*       * Snippet:    11
*       *
*       ****************************************************
*
*!***************************************************************
*
*!
*!      Procedure: _QU70J2FYA
*!
*!      Called by: CASMENU.MPR
*!
*!          Calls: CASENTRY.PRG
*!
*!           Uses: CASEBK.DBF
*!
*!***************************************************************
*
PROCEDURE _qu70j2fya
*
* initialize selection specific variables
*
STORE 'GA' TO mtask_id
STORE 'EXEC' TO mtk_id_desc
STORE 'EX' TO mtk_id_type
*
* get the selected task number questions
*
SELECT a.task_no ;
  FROM casebk A ;
  INTO ARRAY tmparray ;
  WHERE a.tk_id_type = 'EX'
*
* save the number of elements for use
*
num_elements = ALEN(tmparray)
*
* do the data entry program
```

```
*
DO casentry


*       ******************************************************
*       *
*       * _QU70J2G35  ON SELECTION BAR 1 OF POPUP ecicasconv
*       *
*       * Procedure Origin:
*       *
*       * From Menu:  CASMENU.MPR,           Record:  100
*       * Called By:  ON SELECTION BAR 1 OF POPUP ecicasconv
*       * Prompt:     CAS Master Database
*       * Snippet:    12
*       *
*       ******************************************************
*
*!*********************************************************************
*
*!
*!      Procedure: _QU70J2G35
*!
*!      Called by: CASMENU.MPR
*!
*!          Calls: CASCONV.PRG
*!
*!*********************************************************************
*
PROCEDURE _qu70j2g35
dconvert = "casdata"
DO casconv WITH dconvert


*       ******************************************************
*       *
*       * _QU70J2G4O  ON SELECTION BAR 2 OF POPUP ecicasconv
*       *
*       * Procedure Origin:
*       *
*       * From Menu:  CASMENU.MPR,           Record:  101
*       * Called By:  ON SELECTION BAR 2 OF POPUP ecicasconv
*       * Prompt:     CAS Outcome Database
*       * Snippet:    13
*       *
*        ******************************************************
*
*!************************************************************************
*
*!
*!      Procedure: _QU70J2G4O
*!
*!      Called by: CASMENU.MPR
*!
*!         Calls: CASCONV.PRG
*!
*!************************************************************************
*
PROCEDURE _qu70j2g4o
dconvert = "casoutc"
DO casconv WITH dconvert
*: EOF: CASMENU.MPR
```

APPENDIX J

PROGRAM TO ENTER DATA INTO CAS DATABASE

1. Program: casentry.prg

2. Author: D.Butterfield, PRC Inc.

3. Date: 30 March 1994

4. Notes: Program produced to provide the user with a data entry screen which resembles the Close Air Support data collection forms/books. The user can open a book and proceed page by page entering the data. Screen representation emulates book form layout.

5. Usage: Program is called from the casproj.app Close Air Support menu item by selecting Data Entry - CAS Data.

6. Files: Uses the casdata.dbf database to store entered data.

7. Problems: User can not alter previous page data.

8. History:

Date	Name	Ver	Modifications	By
03/30/94	casentry	1.0	original	dbb
07/21/94	casentry	1.1	no detail	dbb

```
*******************************************************************

#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

* if an error occurs inform the user
*ON ERROR DO caserr WITH ERROR(), MESSAGE()

* set the noise off
SET BELL OFF

* open files non-exclusively
SET EXCLUSIVE OFF

* reprocessing of unsuccessful locks is automatic
* reprocessing of unsuccessful locks is automatic
SET REPROCESS TO AUTOMATIC

IF SET("TALK") = "ON"
          SET TALK OFF
          m.talkstat = "ON"
ELSE
          m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

m.rborder = SET("READBORDER")
SET READBORDER ON

m.currarea = SELECT()

*
*        CAS/Windows Databases, Indexes, Relations
*

external array tmparray
*
* initialize the task_array with task_no and values
*
dimension task_array(num_elements, 2)
for task_ptr = 1 to num_elements step 1
          store tmparray(task_ptr) to task_array(task_ptr, 1)
          if (right(trim(tmparray(task_ptr)), 3) != 'REM')
                    store 8 to task_array(task_ptr, 2)
          else
                    store "" to task_array(task_ptr, 2)
          endif
endfor

*
* initialize the data database for use
*
current_dbase = "casdata.dbf"

IF USED(current_dbase)
          SELECT current_dbase
          SET ORDER TO 0
ELSE
          SELECT 0
          USE (LOCFILE(current_dbase,"DBF","Where is current_dbase?"));
                    AGAIN ALIAS current_dbase ;
                    ORDER 0
ENDIF

SET ORDER TO 0

*
* store the task_id variables
*
store "    " to mrotation
store "        " to mtrng_day
store "    " to mtime
store "                " to munit_obs
store "     " to moc_cs
store "                " to mmission
store "    " to mcas_mis
store "       " to mtask_no
store 0 to mscore

*
* Windows Window definitions
*
IF NOT WEXIST("_cas_entry")
          DEFINE WINDOW _cas_entry ;
                    AT 0.000, 0.000 ;
                    SIZE 28.615,98.400 ;
                    FONT "MS Sans Serif", 8 ;
                    TITLE "Close Air Support Data Entry" ;
                    FLOAT ;
                    CLOSE ;
                    MINIMIZE ;
                    SYSTEM
          MOVE WINDOW _cas_entry CENTER
ENDIF

*
*              CAS/Windows Screen Layout
*

#REGION 1
IF WVISIBLE("_cas_entry")
          ACTIVATE WINDOW _cas_entry SAME
ELSE
          ACTIVATE WINDOW _cas_entry NOSHOW
```

```
ENDIF

IF NOT WVISIBLE("_cas_entry")
        ACTIVATE WINDOW _cas_entry
ENDIF

*
* Prompts and input common to each data input form
*
* entry screen line #1
*
@ 0.538,1.400 SAY "Rotation: "  ;
        FONT "MS Sans Serif", 8 ;
        STYLE "BT"
@ 0.538,13.800 GET mrotation ;
        SIZE 1.000,5.000 ;
        DEFAULT " " ;
        FONT "MS Sans Serif", 8 ;
        PICTURE "!99!"

@ 0.538,23.200 SAY "Trng Day: "  ;
        FONT "MS Sans Serif", 8 ;
        STYLE "BT"
@ 0.538,36.000 GET mtrng_day ;
        SIZE 1.000,8.000 ;
        DEFAULT " " ;
        FONT "MS Sans Serif", 8 ;
        PICTURE "@!"

@ 0.538,48.400 SAY "Time:"  ;
        FONT "MS Sans Serif", 8 ;
        STYLE "BT"
@ 0.538,56.400 GET mtime ;
        SIZE 1.000,5.000 ;
        DEFAULT " " ;
        RANGE "0001","2400" ;
        FONT "MS Sans Serif", 8 ;
        PICTURE "9999"

@ 0.538,66.000 SAY "Echelon: "  ;
        FONT "MS Sans Serif", 8 ;
        STYLE "BT"
@ 0.325,77.600 GET mechelon ;
        PICTURE "@^ Battalion;Task Force;Brigade;Division;Corps;Company" ;
        SIZE 1.538,16.000 ;
```

```
            DEFAULT "Battalion" ;
            FONT "MS Sans Serif", 8 ;
            STYLE "B"
*
* entry screen line #2
*
@ 2.000,1.600 SAY "Unit Observed: "  ;
            FONT "MS Sans Serif", 8 ;
            STYLE "BT"
@ 2.000,21.200 GET munit_obs ;
            SIZE 1.000,23.000 ;
            DEFAULT " " ;
            FONT "MS Sans Serif", 8 ;
            PICTURE "@!"

@ 2.000,61.200 SAY "O/C Callsign: "  ;
            FONT "MS Sans Serif", 8 ;
            STYLE "BT"
@ 2.000,78.200 GET moc_cs ;
            SIZE 1.000,7.000 ;
            DEFAULT " " ;
            FONT "MS Sans Serif", 8 ;
            PICTURE "@!"

@ 3.385,1.600 SAY "Mission: "  ;
            FONT "MS Sans Serif", 8 ;
            STYLE "BT"
@ 3.385,21.200 GET mmission ;
            SIZE 1.000,23.000 ;
            DEFAULT " " ;
            FONT "MS Sans Serif", 8 ;
            PICTURE "@!"

@ 3.385,60.000 SAY "CAS Mission#: "  ;
            FONT "MS Sans Serif", 8 ;
            STYLE "BT"
@ 3.385,78.200 GET mcas_mis ;
            SIZE 1.000,2.500 ;
            DEFAULT 0 ;
            FONT "MS Sans Serif", 8 ;
            PICTURE "99"

@ 4.846,1.600 TO 4.846,96.800 ;
            PEN 1, 8 ;
            STYLE "1"
```

```
@ 5.154,1.400 SAY "Task "  ;
          FONT "MS Sans Serif", 8 ;
          STYLE "BT"
@ 5.154,24.500 SAY "Measure"  ;
          FONT "MS Sans Serif", 8 ;
          STYLE "BT"

READ DEACTIVATE _win_lower()

*
* set initial variables
*
x = 6.615
x_org = 6.615
y_col1 = 2.000
y_col2 = 27.000
y_col3 = y_col1 + 40.000
y_col4 = y_col1 + 65.000
y1 = 2.000
y2 = 27.000
field_ptr = 11
num_fields = fcount()
column_limit = 20

*
* step through task numbers and obtain the input data
*
task_ptr = 1
do while task_ptr <= num_elements
          do while right(trim(task_array(task_ptr, 1)), 3) <> "REM"
                    @ x,y1 SAY task_array(task_ptr, 1);
                    SIZE 1.000,14.000 ;
                    FONT "MS Sans Serif", 8

                    @ x,y2 GET task_array(task_ptr, 2) ;
                    SIZE 1.000,2.000 ;
                    DEFAULT 0 ;
                    RANGE 1,8 ;
                    FONT "MS Sans Serif", 8 ;
                    PICTURE "@K"

                    if x >= column_limit
                              x = x_org
                              y1 = y_col3
                              y2 = y_col4
```

```
        else
                x = x + 1
        endif
        task_ptr = task_ptr + 1
    enddo

    *
    * Input the remarks from the form
    *
    @ 24.000,2.000 EDIT task_array(task_ptr, 2);
    SIZE 4.150,93.600,0.000 ;
    PICTURE "@K" ;
    DEFAULT " " ;
    FONT "MS Sans Serif", 8 ;
    SCROLL

    @ 22.308,1.800 SAY "Remarks: (Enter a TAB to end and save remarks)" ;
    FONT "MS Sans Serif", 8 ;
    STYLE "BT"

    @ 21.769,5.200 TO 21.846,5.200 ;
    PEN 1, 8

    *
    * Next Page Button code
    *
    @ 21.692,82.000 GET next_page ;
    PICTURE "@*HT \<Next Page" ;
    SIZE 1.615,11.000,0.500 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B"

    *
    * inititate the read for a page of data
    *
    READ DEACTIVATE _win_lower()

    task_ptr = task_ptr + 1
    x = x_org
    y1 = y_col1
    y2 = y_col2
    @ 6.500,1 CLEAR TO 21.890,80
enddo
```

```
*
* get the echelon abbreviation for the database
*
do case
        case mechelon = "Battalion"
                store "BN" to mselech
        case mechelon = "Task Force"
                store "TF" to mselech
        case mechelon = "Brigade"
                store "BDE" to mselech
        case mechelon = "Division"
                store "DIV" to mselech
        case mechelon = "CORPS"
                store "CORPS" to mselech
        case mechelon = "Company"
                store "COMP" to mselech
endcase

*
* move the newly entered data from the array to a new record
*
for task_ptr = 1 to num_elements step 1
        append blank
        replace rotation with mrotation
        replace trng_day with mtrng_day
        replace time with mtime
        replace echelon with mselech
        replace unit_obs with munit_obs
        replace oc_cs with moc_cs
        replace mission with mmission
        replace cas_mis with mcas_mis
        replace task_id WITH mtask_id
        replace task_no with task_array(task_ptr, 1)

        if (right(trim(task_no), 3) != 'REM')
                replace score with task_array(task_ptr, 2)
        else
                replace score with 0
                replace remarks with task_array(task_ptr, 2)
        endif
endfor

RELEASE WINDOW _cas_entry

*
```

```
*               Windows Closing Databases
*

IF USED(current_dbase)
          SELECT current_dbase
          USE
ENDIF

SELECT (m.currarea)

#REGION 0

SET READBORDER &rborder

IF m.talkstat = "ON"
          SET TALK ON
ENDIF
IF m.compstat = "ON"
          SET COMPATIBLE ON
ENDIF

*
* close all databases in use so next input screen will
* not find difficulty in opening like databases
*
CLOSE DATABASES

* reset error routine to default
ON ERROR

*
* allow another window to overlay data entry window
* do not terminate data entry 'read'
*
FUNCTION _win_lower
RETURN .F.
```

<u>APPENDIX K</u>

PROGRAM TO ENTER DATA INTO THE CAS OUTCOME DATABASE

1. Program: casout1.prg

2. Author: D.Butterfield, PRC Inc.

3. Date: 12 May 1994

4. Notes: Program produced to provide the user with a data entry screen for use in entering the CAS OUTCOME data.

5. Usage: Program is called from the casproj.app menu selection under Close Air Support. select Data Entry - Outcome.

6. Files: Uses the casoutc.dbf database to store entered data.

7. Problems: None noted.

8. History:

Date	Name	Ver	Modifications	by
04/12/94	casout1	1.0	original	dbb
07/21/94	casout1	1.1	no detail	dbb

```
*****************************************************************

#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

SET BELL OFF

SET EXCLUSIVE OFF

SET REPROCESS TO AUTOMATIC

IF SET("TALK") = "ON"
        SET TALK OFF
```

```
          m.talkstat = "ON"
ELSE
          m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

m.rborder = SET("READBORDER")
SET READBORDER ON

m.currarea = SELECT()

mrotation = ""
mmission = ""
mdtg = ""
moc_cs = ""
mleth_a = 0
mleth_b = 0
msurv_a = 0
msurv_b = 0
mcom_tro = 0
mcom_ter = 0
mcom_mis = 0
mcom_ene = 0
mcom_tim = 0

valid_data = .T.

*       *****************************************************
*       *
*       *     CASOUT1/Windows Databases, Indexes, Relations
*       *
*       *****************************************************
*

IF USED("casoutc")
          SELECT casoutc
          SET ORDER TO 0
ELSE
          SELECT 0
          USE (LOCFILE("casoutc.dbf","DBF","Where is casoutc?"));
                    AGAIN ALIAS casoutc ;
                    ORDER 0
ENDIF
```

```
*       ******************************************************
*       *
*       *             Windows Window definitions
*       *
*       ******************************************************
*

IF NOT WEXIST("_qpy0qq5w5")
        DEFINE WINDOW _qpy0qq5w5 ;
                AT 0.000, 17.000 ;
                SIZE 29.000,71.200 ;
                FONT "MS Sans Serif", 8 ;
                FLOAT ;
                NOCLOSE ;
                MINIMIZE ;
                SYSTEM
ENDIF

IF NOT WEXIST("_qpy0qq5zi")
        DEFINE WINDOW _qpy0qq5zi ;
                AT 0.500, 23.000 ;
                SIZE 29.923,73.400 ;
                FONT "MS Sans Serif", 8 ;
                FLOAT ;
                NOCLOSE ;
                MINIMIZE ;
                SYSTEM
ENDIF


*       ******************************************************
*       *
*       *            CASOUT1/Windows Screen Layout
*       *
*       ******************************************************
*

#REGION 1
IF WVISIBLE("_qpy0qq5w5")
        ACTIVATE WINDOW _qpy0qq5w5 SAME
ELSE
        ACTIVATE WINDOW _qpy0qq5w5 NOSHOW
ENDIF
@ 0.538,1.200 SAY "Rotation: "  ;
        FONT "MS Sans Serif", 8 ;
```

```
          STYLE "BT"
@ 0.538,38.200 SAY "Mission: "  ;
          FONT "MS Sans Serif", 8 ;
          STYLE "BT"
@ 3.385,1.800 TO 3.385,68.800 ;
          PEN 1, 8 ;
          STYLE "1"
@ 21.769,5.200 TO 21.846,5.200 ;
          PEN 1, 8
@ 1.846,5.600 SAY "DTG:"  ;
          FONT "MS Sans Serif", 8 ;
          STYLE "BT"
@ 4.231,1.200 SAY "Leathality Component"  ;
          FONT "MS Sans Serif", 10 ;
          STYLE "BT"
@ 6.462,3.400 SAY "A: # of Weapons Used"  ;
          FONT "MS Sans Serif", 8 ;
          STYLE "T"
@ 8.462,3.400 SAY "B: # of Vehicles Killed"  ;
          FONT "MS Sans Serif", 8 ;
          STYLE "T"
@ 11.923,1.200 SAY "Survivability Component"  ;
          FONT "MS Sans Serif", 10 ;
          STYLE "BT"
@ 14.154,3.400 SAY "A: # of Aircraft Starting Mission"  ;
          FONT "MS Sans Serif", 8 ;
          STYLE "T"
@ 16.308,3.400 SAY "B: # of Aircraft at the End of Mission"  ;
          FONT "MS Sans Serif", 8 ;
          STYLE "T"
@ 1.846,32.600 SAY "O/C Callsign:"  ;
          FONT "MS Sans Serif", 8 ;
          STYLE "BT"
@ 0.538,13.400 GET mrotation ;
          SIZE 1.000,5.000 ;
          DEFAULT " " ;
          FONT "MS Sans Serif", 8 ;
          PICTURE "!99!"
@ 0.538,48.600 GET mmission ;
          SIZE 1.000,20.200 ;
          DEFAULT " " ;
          FONT "MS Sans Serif", 8 ;
          PICTURE "@!"
@ 1.846,13.400 GET mdtg ;
          SIZE 1.000,16.600 ;
```

```
        DEFAULT " " ;
        FONT "MS Sans Serif", 8 ;
        PICTURE "999999Z !!! 99"
@ 1.846,48.600 GET moc_cs ;
        SIZE 1.000,8.800 ;
        DEFAULT " " ;
        FONT "MS Sans Serif", 8 ;
        PICTURE "@!"
@ 6.385,48.600 GET mleth_a ;
        SIZE 1.000,2.800 ;
        DEFAULT " " ;
        FONT "MS Sans Serif", 8 ;
        PICTURE "99"
@ 8.385,48.600 GET mleth_b ;
        SIZE 1.000,2.600 ;
        DEFAULT " " ;
        FONT "MS Sans Serif", 8 ;
        PICTURE "99"
@ 14.077,48.600 GET msurv_a ;
        SIZE 1.000,2.800 ;
        DEFAULT " " ;
        FONT "MS Sans Serif", 8 ;
        PICTURE "99"
@ 16.231,48.600 GET msurv_b ;
        SIZE 1.000,2.800 ;
        DEFAULT " " ;
        FONT "MS Sans Serif", 8 ;
        PICTURE "99"
@ 26.769,53.800 GET mnext ;
        PICTURE "@*HT \<Next Page" ;
        SIZE 1.615,12.833,0.500 ;
        DEFAULT 1 ;
        FONT "MS Sans Serif", 8 ;
        STYLE "B"

IF NOT WVISIBLE("_qpy0qq5zi")
        ACTIVATE WINDOW _qpy0qq5zi
ENDIF

READ DEACTIVATE _win_lower()

*       ****************************************************
*       *
*       *               CASOUT2/Windows Screen Layout
*       *
*       ******************************************************
*

#REGION 2
IF WVISIBLE("_qpy0qq5zi")
        ACTIVATE WINDOW _qpy0qq5zi SAME
ELSE
        ACTIVATE WINDOW _qpy0qq5zi NOSHOW
ENDIF
@ 0.692,2.400 SAY "Rotation: "  ;
        FONT "MS Sans Serif", 8 ;
        STYLE "BT"
@ 0.692,39.400 SAY "Mission: "  ;
        FONT "MS Sans Serif", 8 ;
        STYLE "BT"
@ 3.231,3.000 TO 3.231,70.000 ;
        PEN 1, 8 ;
        STYLE "1"
@ 21.923,6.400 TO 22.000,6.400 ;
        PEN 1, 8
@ 2.000,6.800 SAY "DTG:"  ;
        FONT "MS Sans Serif", 8 ;
        STYLE "BT"
@ 14.308,2.400 SAY "Debrief Notes"  ;
        FONT "MS Sans Serif", 8 ;
        STYLE "T"
@ 19.154,2.400 SAY "Debrief Notes"  ;
        FONT "MS Sans Serif", 8 ;
        STYLE "T"
@ 2.000,33.800 SAY "O/C Callsign:"  ;
        FONT "MS Sans Serif", 8 ;
        STYLE "BT"
@ 13.308,31.200 SAY "1 = Yes, 0 = No"  ;
        FONT "MS Sans Serif", 8 ;
        STYLE "T"
@ 18.154,31.200 SAY "1 = Yes, 0 = No"  ;
        FONT "MS Sans Serif", 8 ;
        STYLE "T"
@ 22.923,31.200 SAY "1 = Yes, 0 = No"  ;
        FONT "MS Sans Serif", 8 ;
        STYLE "T"
@ 23.923,2.400 SAY "Debrief Notes"  ;
        FONT "MS Sans Serif", 8 ;
        STYLE "T"
@ 9.538,2.400 SAY "Debrief Notes"  ;
```

```
            FONT "MS Sans Serif", 8 ;
            STYLE "T"
@ 4.769,2.400 SAY "Debrief Notes"  ;
            FONT "MS Sans Serif", 8 ;
            STYLE "T"
@ 3.769,31.200 SAY "1 = Yes, 0 = No"  ;
            FONT "MS Sans Serif", 8 ;
            STYLE "T"
@ 8.538,31.200 SAY "1 = Yes, 0 = No"  ;
            FONT "MS Sans Serif", 8 ;
            STYLE "T"
@ 0.692,14.600 SAY mrotation ;
            SIZE 1.000,5.000 ;
            FONT "MS Sans Serif", 8
@ 0.692,50.000 SAY mmission ;
            SIZE 1.000,20.200 ;
            FONT "MS Sans Serif", 8
@ 2.000,14.600 SAY mdtg ;
            SIZE 1.000,16.600 ;
            FONT "MS Sans Serif", 8
@ 2.000,50.000 SAY moc_cs ;
            SIZE 1.000,6.600 ;
            FONT "MS Sans Serif", 8

@ 3.692,50.000 GET mcom_mis ;
            SIZE 1.000,2.800 ;
            DEFAULT " " ;
            FONT "MS Sans Serif", 8 ;
            PICTURE "9";
            RANGE 0, 1
@ 6.000,2.800 EDIT mrem_mis ;
            SIZE 2.000,67.600,0.000 ;
            PICTURE "@!" ;
            DEFAULT "" ;
            FONT "MS Sans Serif", 8 ;
            SCROLL

@ 8.462,50.000 GET mcom_ene ;
            SIZE 1.000,2.800 ;
            DEFAULT " " ;
            FONT "MS Sans Serif", 8 ;
            PICTURE "9";
            RANGE 0, 1
@ 10.769,2.800 EDIT mrem_ene ;
            SIZE 2.000,67.600,0.000 ;
```

```
          PICTURE "@!" ;
          DEFAULT "" ;
          FONT "MS Sans Serif", 8 ;
          SCROLL

@ 13.231,50.000 GET mcom_tro ;
          SIZE 1.000,3.000 ;
          DEFAULT " " ;
          FONT "MS Sans Serif", 8 ;
          PICTURE "9";
          RANGE 0, 1
@ 15.538,2.800 EDIT mrem_tro ;
          SIZE 2.000,67.600,0.000 ;
          PICTURE "@!" ;
          DEFAULT "" ;
          FONT "MS Sans Serif", 8 ;
          SCROLL

@ 18.077,50.000 GET mcom_ter ;
          SIZE 1.000,2.800 ;
          DEFAULT " " ;
          FONT "MS Sans Serif", 8 ;
          PICTURE "9";
          RANGE 0, 1
@ 20.385,2.800 EDIT mrem_ter ;
          SIZE 2.000,67.600,0.000 ;
          PICTURE "@!" ;
          DEFAULT "" ;
          FONT "MS Sans Serif", 8 ;
          SCROLL

@ 22.846,50.000 GET mcom_tim ;
          SIZE 1.000,2.800 ;
          DEFAULT " " ;
          FONT "MS Sans Serif", 8 ;
          PICTURE "9";
          RANGE 0, 1
@ 25.154,2.800 EDIT mrem_tim ;
          SIZE 2.000,67.600,0.000 ;
          PICTURE "@!" ;
          DEFAULT "" ;
          FONT "MS Sans Serif", 8 ;
          SCROLL

@ 27.769,55.000 GET mfinished ;
```

```
          PICTURE "@*HT \<Finished" ;
          SIZE 1.615,12.833,0.500 ;
          DEFAULT 1 ;
          FONT "MS Sans Serif", 8 ;
          STYLE "B"
@  3.769,2.400 SAY "MISSION"  ;
          FONT "MS Sans Serif", 8 ;
          STYLE "BT"
@  8.538,2.400 SAY "ENEMY"  ;
          FONT "MS Sans Serif", 8 ;
          STYLE "BT"
@ 13.308,2.400 SAY "TROOPS"  ;
          FONT "MS Sans Serif", 8 ;
          STYLE "BT"
@ 18.000,2.400 SAY "TERRAIN"  ;
          FONT "MS Sans Serif", 8 ;
          STYLE "BT"
@ 22.923,2.400 SAY "TIME"  ;
          FONT "MS Sans Serif", 8 ;
          STYLE "BT"

IF NOT WVISIBLE("_qpy0qq5w5")
          ACTIVATE WINDOW _qpy0qq5w5
ENDIF

READ DEACTIVATE _win_lower()

if valid_data
          append blank
          replace rotation with mrotation
          replace mission with mmission
          replace dtg with mdtg
          replace oc_cs with moc_cs
          replace leth_a with mleth_a
          replace leth_b with mleth_b
          replace surv_a with msurv_a
          replace surv_b with msurv_b
          replace com_tro with mcom_tro
          replace com_ter with mcom_ter
          replace com_mis with mcom_mis
          replace com_ene with mcom_ene
          replace com_tim with mcom_tim
          replace rem_tro with mrem_tro
          replace rem_ter with mrem_ter
          replace rem_mis with mrem_mis
          replace rem_ene with mrem_ene
          replace rem_tim with mrem_tim
endif

RELEASE WINDOW _qpy0qq5w5
RELEASE WINDOW _qpy0qq5zi

*       ******************************************************
*       *
*       *              Windows Closing Databases
*       *
*       ******************************************************
*

IF USED("casoutc")
          SELECT casoutc
          USE
ENDIF

SELECT (m.currarea)


#REGION 0

SET READBORDER &rborder

IF m.talkstat = "ON"
          SET TALK ON
ENDIF
IF m.compstat = "ON"
          SET COMPATIBLE ON
ENDIF

FUNCTION _win_lower
RETURN .F.

PROCEDURE _invalid_data
          valid_data = .F.
RETURN
```

APPENDIX L

PROGRAM TO ALLOW USER INTERFACE TO SQL QUERY AND REPORTS

1. Program: casrpts.prg

2. Author: D.Butterfield/Jerry Fargo, PRC Inc.

3. Date: 16 April 1994

4. Notes: Program produced to provide the user with desired database data selection criteria and type of report to generate. Once the data and report type are selected from the dropdown menus, the user may preview or print the report.

5. Usage: Program is called from the initial menu for report selection and generation. Titles are displayed over each of the respective dropdown menus. Select items from the menus and then preview or print the report.

6. Files: Uses almost all cas database tables.

7. Problems: None noted.

8. History:

date	name	ver	modifications	by
04/16/94	casrpts	1.0	original	dbb
07/21/94	casrpts	1.1	no detail	dbb
08/02/94	casrpts put into tables for faster execution	1.2	dropdown menus	dbb
10/06/94	casrpts rotations. removed one report	1.3	modified for 'All'	dbb

9. Last modified: 11/18/94 at 12:24:34

10. Procedures and Functions: _CHKFORUPDATE
    _WAITMSG
    _ROTATION
    _MISSION
    _TRAINING

10.	Procedures and Functions:	_CHKFORUPDATE	
		_WAITMSG	
		_ROTATION	
		_MISSION	
		_TRAINING	
		_CALLSIGN	
		_UNITOBS	
		_CREATE_RPT()	
		_STOP_LOOP()	
		_GET_RPT	
		_GET_BOOK	
		_GET_TYPE	
		_GET_ECHLEV	
11.	Set by:	_QU70J2FCT	(procedure in CASMENU.MPR)
12.	Calls:	CASERR.PRG	
		_CHKFORUPDATE	(procedure in CASRPTS.PRG)
		_WAITMSG	(procedure in CASRPTS.PRG)
		_ROTATION	(procedure in CASRPTS.PRG)
		_MISSION	(procedure in CASRPTS.PRG)
		_TRAINING	(procedure in CASRPTS.PRG)
		_CALLSIGN	(procedure in CASRPTS.PRG)
		_UNITOBS	(procedure in CASRPTS.PRG)
		_CREATE_RPT()	(function in CASRPTS.PRG)
		_STOP_LOOP()	(function in CASRPTS.PRG)

```
*********************************************************************

* if an error occurs, inform the user
ON ERROR DO caserr WITH ERROR(), MESSAGE()

* open files non-exclusively
SET EXCLUSIVE OFF

* reprocessing of unsuccessful locks is automatic
SET REPROCESS TO AUTOMATIC

* set the file update prompting off
SET SAFETY OFF
*
* check for updates to database records
*
update_db = .F.
DO _chkforupdate WITH update_db
```

```
IF update_db
   showmsg = .T.
   DO _waitmsg WITH showmsg
ENDIF

*
* initialize arrays at minimum
*
DIMENSION rota_array[1,1]
DIMENSION miss_array[1,1]
DIMENSION trng_array[1,1]
DIMENSION occs_array[1,1]
DIMENSION unit_array[1,1]

*
* make the selection arrays
*
DO _rotation WITH rota_array
DO _mission  WITH miss_array
DO _training WITH trng_array
DO _callsign WITH occs_array
DO _unitobs  WITH unit_array

IF update_db
   DO _waitmsg WITH showmsg
ENDIF

*
* set up the initial program parameters
*
#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
   SET TALK OFF
   m.talkstat = "ON"
ELSE
   m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS
SET EXCLUSIVE OFF
m.rborder = SET("READBORDER")
SET readborder ON
```

```
m.currarea = SELECT()

*
* Windows Databases, Indexes, Relations
*

IF USED("casdata")
   SELECT casdata
   SET ORDER TO 0
ELSE
   SELECT 0
   USE (LOCFILE("casdata.dbf","DBF","Where is casdata?"));
      AGAIN ALIAS casdata ;
      ORDER 0
ENDIF

*
* Windows Window definitions
*
IF NOT WEXIST("_cas_select")
   DEFINE WINDOW _cas_select ;
      AT  0.000, 0.000  ;
      SIZE 29.077,100.200 ;
      FONT "MS Sans Serif", 8 ;
      TITLE "Close Air Support Reports" ;
      FLOAT ;
      CLOSE ;
      MINIMIZE ;
      SYSTEM
   MOVE WINDOW _cas_select CENTER
ENDIF

*
* casrpts/Windows Screen Layout
*
#REGION 1
IF WVISIBLE("_cas_select")
   ACTIVATE WINDOW _cas_select SAME
ELSE
   ACTIVATE WINDOW _cas_select NOSHOW
ENDIF

*
* start the the selection screen loop
*
```

```
dont_stop = .T.
DO WHILE dont_stop
  *
  * selection area titles
  *
  @ 0.615,1.200 SAY "REPORT SELECTIONS"  ;
    FONT "MS Sans Serif", 12 ;
    STYLE "BT"

  @ 0.615,50.400 SAY "QUERY SELECTIONS"  ;
    FONT "MS Sans Serif", 12 ;
    STYLE "BT"
  *
  * selection area boxes
  *
  @ 2.538,1.000 TO 25.692,47.200 ;
    PEN 2, 8

  @ 2.538,50.200 TO 25.692,96.400 ;
    PEN 2, 8
  *
  * left side list selection titles
  *
  @ 3.385,3.600 SAY "Title Selection"  ;
    FONT "MS Sans Serif", 8 ;
    STYLE "T"

  @ 8.077,3.600 SAY "Summary Selection"  ;
    FONT "MS Sans Serif", 8 ;
    STYLE "T"

  @ 12.769,3.600 SAY "Task Type Selection"  ;
    FONT "MS Sans Serif", 8 ;
    STYLE "T"

  @ 17.308,3.600 SAY "Task Plan/Prep Selection"  ;
    FONT "MS Sans Serif", 8 ;
    STYLE "T"

  @ 21.615,3.600 SAY "Echelon Level Selection"  ;
    FONT "MS Sans Serif", 8 ;
    STYLE "T"
  *
  * left side selection gets and lists
  *
```

```
@ 4.923,3.600 GET mtitle ;
  PICTURE "@^ Task Assessment Distribution;Task Remarks Comparison" ;
  SIZE 1.538,34.167 ;
  DEFAULT "Task Assessment Distribution" ;
  FONT "MS Sans Serif", 8 ;
  STYLE "B"

@ 9.538,3.600 GET msum_title ;
  PICTURE "@^ Training Day;Mission;Rotation;Training Center" ;
  SIZE 1.538,34.167 ;
  DEFAULT "Training Day" ;
  FONT "MS Sans Serif", 8 ;
  STYLE "B"

@ 14.154,3.600 GET mtk_title ;
  PICTURE "@^ AFAC;TACP;Maneuver;CAS Execution" ;
  SIZE 1.538,34.167 ;
  DEFAULT "AFAC" ;
  FONT "MS Sans Serif", 8 ;
  STYLE "B"

@ 18.769,3.600 GET mtk_type ;
  PICTURE "@^ All;Planning;Preparation;Execution" ;
  SIZE 1.538,34.000 ;
  DEFAULT "All" ;
  FONT "MS Sans Serif", 8 ;
  STYLE "B"

@ 23.077,3.600 GET mechlev ;
  PICTURE "@^ All;Battalion;Task Force;Brigade;Division;Corps;Company" ;
  SIZE 1.538,34.000 ;
  DEFAULT "All" ;
  FONT "MS Sans Serif", 8 ;
  STYLE "B"

*
* right side selection titles
*
@ 3.385,52.600 SAY "Rotation Number"  ;
  FONT "MS Sans Serif", 8 ;
  STYLE "T"

@ 7.154,52.600 SAY "Unit Observed"  ;
  FONT "MS Sans Serif", 8 ;
  STYLE "T"
```

```
@ 10.846,52.600 SAY "Mission"  ;
  FONT "MS Sans Serif", 8 ;
  STYLE "T"

@ 14.385,52.600 SAY "Training Day"  ;
  FONT "MS Sans Serif", 8 ;
  STYLE "T"

@ 18.000,52.600 SAY "O/C"  ;
  FONT "MS Sans Serif", 8 ;
  STYLE "T"

@ 21.538,52.600 SAY "Report Level"  ;
  FONT "MS Sans Serif", 8 ;
  STYLE "T"
*
* right side selection gets and lists
*
@ 4.923,52.600 GET mrot_num ;
  PICTURE "@^" ;
  FROM rota_array ;
  SIZE 1.538,34.167 ;
  DEFAULT 1 ;
  RANGE 1 ;
  FONT "MS Sans Serif", 8 ;
  STYLE "B"

@ 8.615,52.600 GET munit_obs ;
  PICTURE "@^" ;
  FROM unit_array ;
  SIZE 1.538,34.167 ;
  DEFAULT "All" ;
  FONT "MS Sans Serif", 8 ;
  STYLE "B"

@ 12.231,52.600 GET mmission ;
  PICTURE "@^" ;
  FROM miss_array ;
  SIZE 1.538,34.167 ;
  DEFAULT "All" ;
  FONT "MS Sans Serif", 8 ;
  STYLE "B"

@ 15.769,52.600 GET mtrng_day ;
```

```
    PICTURE "@^" ;
    FROM trng_array ;
    SIZE 1.538,34.167 ;
    DEFAULT "All" ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B"

  @ 19.385,52.600 GET moc_cs ;
    PICTURE "@^" ;
    FROM occs_array ;
    SIZE 1.538,34.167 ;
    DEFAULT "All" ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B"

  @ 23.077,52.600 GET mlevel ;
    PICTURE "@^ Level 1 (M01);Level 2 (M01, M01a);Level 3 (M01, M01a,
M01a1);Level 4 (All Task Numbers)" ;
    SIZE 1.538,34.167 ;
    DEFAULT "Level 1 (M01)" ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B"
  *
  * preview, print and cancel buttons
  *
  mpreview = 0
  mprint = 0
  @ 26.692,12.000 GET mpreview ;
    PICTURE "@*HT Preview" ;
    SIZE 1.769,12.500,0.667 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B" ;
    MESSAGE "Preview the Report." ;
    VALID _create_rpt()

  @ 26.692,41.000 GET mprint ;
    PICTURE "@*HT Print" ;
    SIZE 1.769,12.500,0.667 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B" ;
    MESSAGE "Print the Report." ;
    VALID _create_rpt()
  @ 26.692,70.000 GET mexit ;
    PICTURE "@*HT Cancel" ;
    SIZE 1.769,12.500,0.667 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B" ;
    WHEN _stop_loop()
  *
  * make sure selection window is visible
  *
  IF NOT WVISIBLE("_cas_select")
    ACTIVATE WINDOW _cas_select
  ENDIF
  *
  * read the user selections
  *
  READ CYCLE

  * end of dont_stop loop
ENDDO

*
* close the windows
*
RELEASE WINDOW _cas_select

*
* Windows Closing Databases
*
IF USED("casdata")
  SELECT casdata
  USE
ENDIF

SELECT (m.currarea)

#REGION 0

SET readborder &rborder

IF m.talkstat = "ON"
  SET TALK ON
ENDIF
IF m.compstat = "ON"
  SET COMPATIBLE ON
```

```
ENDIF

* reset the on error routine to the default
ON ERROR

************************* end of program **********************

******************************************************************
* _create_rpt() - create a report for preview or printing
******************************************************************
*!***********************************************************************
*
*!
*!      Function: _CREATE_RPT
*!
*!     Called by: CASRPTS.PRG
*!
*!         Calls: _GET_RPT         (procedure in CASRPTS.PRG)
*!              : _GET_BOOK        (procedure in CASRPTS.PRG)
*!              : _GET_TYPE        (procedure in CASRPTS.PRG)
*!              : _GET_ECHLEV      (procedure in CASRPTS.PRG)
*!              : CASCRPT1.PRG
*!              : CASNONE.PRG
*!              : CASLOC.PRG
*!              : CASCRPT2.PRG
*!              : CASCRPT3.PRG
*!
*!  Report Forms: CASRPT1.FRX
*!              : CASRPT2.FRX
*!              : CASRPT3.FRX
*!
*!***********************************************************************
*
FUNCTION _create_rpt      && mpreview/mprint WHEN
#REGION 1
* initialize local variables
sel_title1 = ""                 && report title
sel_title2 = ""                 && report summary type
sel_title3 = ""                 && report task type (manuever, afac, etc.)
sel_title4 = ""                 && report task type (plan/prep)
sel_title5 = ""                 && report echelon level
sel_rpt = 0                                 && report format to use/do
sel_book = ""                   && task book (maneuver, afac, etc.)
sel_type = ""                   && task type (plan/prep)
sel_tkid = ""                   && task id type (MO, MF, A, G, GA, etc.)
```

```
sel_echlev = ""                                && task echelon level

* construct all of the report title lines from selected
* parameters and then make the queries to obtain the data
* for the reports.
* following put into procedures to reduce this function size
DO _get_rpt WITH sel_rpt, sel_title1, sel_title2
DO _get_book WITH sel_book, sel_title3, sel_tkid
DO _get_type WITH sel_type, sel_title4
DO _get_echlev WITH sel_echlev, sel_title5

*
* task number/id level
*
DO CASE
CASE mlevel = "Level 1 (M01)"
   sel_level = "caslev1"
CASE mlevel = "Level 2 (M01, M01a)"
   sel_level = "caslev2"
CASE mlevel = "Level 3 (M01, M01a, M01a1)"
   sel_level = "caslev3"
CASE mlevel = "Level 4 (All Task Numbers)"
   sel_level = "caslev4"
OTHERWISE
   sel_level = "caslev4"
ENDCASE

*
* save the selected rotation for database search
*
sel_rota = rota_array(mrot_num)

*
* final report headings
*
sel_title6 = "ROTATION: " + sel_rota + ", "
sel_title7 = "UNIT: " + TRIM(munit_obs) + ", "
sel_title8 = "MISSION: " + TRIM(mmission) + ", "
sel_title9 = "TRAINING DAY: " + TRIM(mtrng_day) + ", "
sel_titl10 = "O/C: " + TRIM(moc_cs)
sel_long1 = sel_title6+sel_title7+sel_title8+sel_title9+sel_titl10

*
* do the individual reports according to the selection
* if no records were found, do not generate a report.
* inform the user via a popup window.
* give the user the option of printing to a file or printer
mlocation = 1
mokay = 1
DO CASE
  *
  * task assessment consistency report
  *
CASE sel_rpt = 1
  mrfile = "CASPRN1.TXT"
  DO cascrpt1

  IF RECCOUNT() < 1
    DO casnone
  ELSE
    IF mpreview = 1
      REPORT FORM casrpt1 PREVIEW
    ELSE
      DO casloc WITH mlocation, mokay, mrfile
      IF mokay = 1
        IF mlocation = 1
          SET CONSOLE OFF
          REPORT FORM casrpt1 TO PRINTER
          SET CONSOLE ON
        ELSE
          SET CONSOLE OFF
          REPORT FORM casrpt1 TO FILE (mrfile)
          SET CONSOLE ON
        ENDIF
      ENDIF
    ENDIF
  ENDIF

  *
  * task assessment distribution report
  *
CASE sel_rpt = 2
  mrfile = "CASPRN2.TXT"
  DO cascrpt2

  IF RECCOUNT() < 1
    DO casnone
  ELSE
    IF mpreview = 1
      REPORT FORM casrpt2 PREVIEW
      ELSE
        DO casloc WITH mlocation, mokay, mrfile
        IF mokay = 1
          IF mlocation = 1
            SET CONSOLE OFF
            REPORT FORM casrpt2 TO PRINTER
            SET CONSOLE ON
          ELSE
            SET CONSOLE OFF
            REPORT FORM casrpt2 TO FILE (mrfile)
            SET CONSOLE ON
          ENDIF
        ENDIF
      ENDIF
    ENDIF
    *
    * task remarks comparison report
    *
  CASE sel_rpt = 3
    mrfile = "CASPRN3.TXT"
    DO cascrpt3

    IF RECCOUNT() < 1
      DO casnone
    ELSE
      IF mpreview = 1
        REPORT FORM casrpt3 PREVIEW
      ELSE
        DO casloc WITH mlocation, mokay, mrfile
        IF mokay = 1
          IF mlocation = 1
            SET CONSOLE OFF
            REPORT FORM casrpt3 TO PRINTER
            SET CONSOLE ON
          ELSE
            SET CONSOLE OFF
            REPORT FORM casrpt3 TO FILE (mrfile)
            SET CONSOLE ON
          ENDIF
        ENDIF
      ENDIF
    ENDIF

  ENDCASE
```

```
RETURN

*********************************************************************
* _chkforupdate() - check for need to update dropdown menu tables
*********************************************************************
*!*************************************************************************
*
*!
*!      Procedure: _CHKFORUPDATE
*!
*!      Called by: CASRPTS.PRG
*!
*!*************************************************************************
*
PROCEDURE _chkforupdate
PARAMETER update_db
pres_rec_no = 0
* set database
current_db = "casdata.dbf"
IF USED(current_dbase)
   SELECT current_db
ELSE
   USE (current_db)
ENDIF

* assign present record number value
pres_rec_num = RECCOUNT()

* reset current database
current_db = "casrecno.dbf"
* open new database
IF USED(current_db)
   SELECT current_db
ELSE
   USE (current_db)
ENDIF

* compare previous record number value with present
IF pres_rec_num != num_recs
   * not the same, update the dropdown menu tables
   update_db = .T.
   REPLACE num_recs WITH pres_rec_num
ENDIF
* close the open databases
CLOSE DATABASES
```

```
RETURN

*************************************************************
* _waitmsg() - display a user wait message
*************************************************************
*!*************************************************************************
*
*!
*!     Procedure: _WAITMSG
*!
*!     Called by: CASRPTS.PRG
*!
*!*************************************************************************
*
PROCEDURE _waitmsg
PARAMETER showmsg
IF showmsg
  showmsg = .F.

  IF NOT WEXIST("_waitabit")
    DEFINE WINDOW _waitabit ;
      AT  0.000, 0.000  ;
      SIZE 11.154,80.600 ;
      FONT "MS Sans Serif", 8 ;
      FLOAT ;
      CLOSE ;
      NOMINIMIZE
  ENDIF

  IF WVISIBLE("_waitabit")
    ACTIVATE WINDOW _waitabit SAME
  ELSE
    ACTIVATE WINDOW _waitabit NOSHOW
  ENDIF

  @ 4.000,7.800 SAY "   Please wait while I update the menu tables...";
    FONT "MS Sans Serif", 10 ;
    STYLE "BT"

  @ 1.846,1.600 TO 9.384,79.000 ;
    PEN 2, 8

  IF NOT WVISIBLE("_waitabit")
    ACTIVATE WINDOW _waitabit
  ENDIF
```

```
ELSE
  IF WVISIBLE("_waitabit")
    RELEASE WINDOW _waitabit
  ENDIF
ENDIF
RETURN

**********************************************************************
* _rotation() - get available rotations and save in an array
**********************************************************************
*!*************************************************************************
*
*!
*!      Procedure: _ROTATION
*!
*!      Called by: CASRPTS.PRG
*!
*!           Uses: CASDATA.DBF
*!               : CASROTA.DBF
*!
*!*************************************************************************
*
PROCEDURE _rotation
PARAMETER rota_array
* check for table update
IF update_db OR !FILE("casrota.dbf")
  SELECT DISTINCT a.rotation ;
    FROM casdata A ;
    INTO TABLE casrota
ENDIF

* obtain the actual database rotations available
SELECT DISTINCT a.rotation ;
  FROM casrota A ;
  INTO ARRAY temp_array

* insert an 'All' selection into the array list
m.count = _TALLY
DIMENSION temp_array(m.count + 1, 1)
= AINS(temp_array, m.count + 1)
temp_array[m.count+1] = "All"

= ASORT(temp_array)
= ACOPY(temp_array, rota_array)
RETURN
```

```
**********************************************************************
* _mission() - get available missions and save in an array
**********************************************************************
*!**************************************************************************
*
*!
*!      Procedure: _MISSION
*!
*!      Called by: CASRPTS.PRG
*!
*!           Uses: CASDATA.DBF
*!               : CASMISS.DBF
*!
*!**************************************************************************
*
PROCEDURE _mission
PARAMETER miss_array
* check for table update
IF update_db OR !FILE("casmiss.dbf")
   SELECT DISTINCT a.mission ;
      FROM casdata A ;
      INTO TABLE casmiss
ENDIF

* obtain the actual database missions available
SELECT DISTINCT a.mission ;
   FROM casmiss A;
   INTO ARRAY temp_array

* insert an 'All' selection into the array list
m.count = _TALLY
DIMENSION temp_array(m.count + 1, 1)
= AINS(temp_array, m.count + 1)
temp_array[m.count+1] = "All"

= ACOPY(temp_array, miss_array)
RETURN

**********************************************************************
* _training() - get available training days and save in an array
**********************************************************************
*!**************************************************************************
*
*!
*!      Procedure: _TRAINING
```

```
*!
*!    Called by: CASRPTS.PRG
*!
*!         Uses: CASDATA.DBF
*!             : CASTRNG.DBF
*!
*!**************************************************************************
*
PROCEDURE _training
PARAMETER trng_array
* check for table update
IF update_db OR !FILE("castrng.dbf")
   SELECT DISTINCT a.trng_day ;
      FROM casdata A ;
      INTO TABLE castrng
ENDIF

* obtain the actual database training days available
SELECT DISTINCT a.trng_day ;
   FROM castrng A;
   INTO ARRAY temp_array

* insert an 'All' selection into the array list
m.count = _TALLY
DIMENSION temp_array(m.count + 1, 1)
= AINS(temp_array, m.count + 1)
temp_array[m.count+1] = "All"

= ASORT(temp_array)
= ACOPY(temp_array, trng_array)
RETURN

**********************************************************************
* _callsign() - get available O/C callsigns and save in an array
**********************************************************************
*!**************************************************************************
*
*!
*!    Procedure: _CALLSIGN
*!
*!    Called by: CASRPTS.PRG
*!
*!         Uses: CASDATA.DBF
*!             : CASOCCS.DBF
*!
```

```
*!*************************************************************************
*
PROCEDURE _callsign
PARAMETERS occs_array
* check for table update
IF update_db OR !FILE("casoccs.dbf")
   SELECT DISTINCT a.oc_cs ;
      FROM casdata A ;
      INTO TABLE casoccs
ENDIF

* obtain the actual database o/c callsigns available
SELECT DISTINCT a.oc_cs ;
   FROM casoccs A ;
   INTO ARRAY temp_array

* insert an 'All' selection into the array list
m.count = _TALLY
DIMENSION temp_array(m.count + 1, 1)
= AINS(temp_array, m.count + 1)
temp_array[m.count+1] = "All"

= ASORT(temp_array)
= ACOPY(temp_array, occs_array)
RETURN

*******************************************************************
* _unitobs() - get available units and save in an array
*******************************************************************
*!*************************************************************************
*
*!
*!      Procedure: _UNITOBS
*!
*!      Called by: CASRPTS.PRG
*!
*!           Uses: CASDATA.DBF
*!               : CASUNIT.DBF
*!
*!*************************************************************************
*
PROCEDURE _unitobs
PARAMETERS unit_array
* check for table update
IF update_db OR !FILE("casunit.dbf")
```

```
  SELECT DISTINCT a.unit_obs ;
    FROM casdata A ;
    INTO TABLE casunit
ENDIF

* obtain the actual database units available
SELECT DISTINCT a.unit_obs ;
  FROM casunit A ;
  INTO ARRAY temp_array

* insert an 'All' selection into the array list
m.count = _TALLY
DIMENSION temp_array(m.count + 1, 1)
= AINS(temp_array, m.count + 1)
temp_array[m.count+1] = "All"

= ASORT(temp_array)
= ACOPY(temp_array, unit_array)
RETURN

*******************************************************************
* _get_rpt() - assign the title lines and report form number
*******************************************************************
*!*****************************************************************************
*
*!
*!      Procedure: _GET_RPT
*!
*!      Called by: _CREATE_RPT()       (function in CASRPTS.PRG)
*!
*!*****************************************************************************
*
PROCEDURE _get_rpt
PARAMETER sel_rpt, sel_title1, sel_title2
*
* report title line one, item one
*
DO CASE
CASE mtitle = "Task Assessment Consistency"
  sel_title1 = "TASK ASSESSMENT CONSISTENCY"
  sel_rpt = 1
CASE mtitle = "Task Assessment Distribution"
  sel_title1 = "TASK ASSESSMENT DISTRIBUTION"
  sel_rpt = 2
CASE mtitle = "Task Remarks Comparison"
```

```
  sel_title1 = "TASK REMARKS COMPARISON"
  sel_rpt = 3
ENDCASE

*
* report title line one, item two
*
DO CASE
CASE msum_title = "Training Day"
  sel_title2 = ", TRAINING DAY SUMMARY"
CASE msum_title = "Mission"
  sel_title2 = ", MISSION SUMMARY"
CASE msum_title = "Rotation"
  sel_title2 = ", ROTATION SUMMARY"
CASE msum_title = "Training Center"
  sel_title2 = ", TRAINING CENTER SUMMARY"
ENDCASE
RETURN

********************************************************************
* _get_book() - assign title line three, cas book and task_id
********************************************************************
*!*************************************************************************
*
*!
*!      Procedure: _GET_BOOK
*!
*!      Called by: _CREATE_RPT()      (function in CASRPTS.PRG)
*!
*!*************************************************************************
*
PROCEDURE _get_book
PARAMETER sel_book, sel_title3, sel_tkid
* report title line two, item one
* use task_no list from task id books
DO CASE
CASE mtk_title = "AFAC"
  sel_title3 = "AFAC "
  sel_book = "casabk"
  sel_tkid = "A"
CASE mtk_title = "TACP"
  sel_title3 = "TACP "
  sel_book = "castbk"
  sel_tkid = "G"
CASE mtk_title = "Maneuver"
```

```
  sel_title3 = "MANEUVER "
  sel_book = "casmbk"
  sel_tkid = "M"
CASE mtk_title = "CAS Execution"
  sel_title3 = ""
  sel_book = "casebk"
  sel_tkid = "GA"
ENDCASE
RETURN

***********************************************************************
* _get_type() - assign title line four and task type
***********************************************************************
*!***********************************************************************
*
*!
*!      Procedure: _GET_TYPE
*!
*!      Called by: _CREATE_RPT()      (function in CASRPTS.PRG)
*!
*!***********************************************************************
*
PROCEDURE _get_type
PARAMETER sel_type, sel_title4
*
* title line two, item two
*
DO CASE
CASE mtk_type = "All"
  sel_title4 = "ALL TASKS"
  sel_type = ""
CASE mtk_type = "Planning"
  sel_title4 = "PLANNING TASKS"
  sel_type = "PL"
CASE mtk_type = "Preparation"
  sel_title4 = "PREPARATION TASKS"
  sel_type = "PR"
CASE mtk_type = "Execution"
  sel_title4 = "EXECUTION TASKS"
  sel_type = "EX"
ENDCASE
RETURN

***********************************************************************
* _get_echlev() - assign title line five and echelon level
```

```
*******************************************************************
*!*********************************************************************
*
*!
*!      Procedure: _GET_ECHLEV
*!
*!      Called by: _CREATE_RPT()      (function in CASRPTS.PRG)
*!
*!*********************************************************************
*
PROCEDURE _get_echlev
PARAMETER sel_echlev, sel_title5
*
* report title line two, item three
*
DO CASE
CASE mechlev = "Battalion"
  sel_title5 = ", BATTALION LEVEL"
  sel_echlev = "BN"
CASE mechlev = "Task Force"
  sel_title5 = ", TASK FORCE LEVEL"
  sel_echlev = "TF"
CASE mechlev = "Brigade"
  sel_title5 = ", BRIGADE LEVEL"
  sel_echlev = "BDE"
CASE mechlev = "Division"
  sel_title5 = ", DIVISION LEVEL"
  sel_echlev = "DIV"
CASE mechlev = "Corps"
  sel_title5 = ", CORPS LEVEL"
  sel_echlev = "CORPS"
CASE mechlev = "Company"
  sel_title5 = ", COMPANY LEVEL"
  sel_echlev = "COMP"
CASE mechlev = "All"
  sel_title5 = ", ALL LEVELS"
  sel_echlev = ""
ENDCASE
RETURN

***************************************************************
* _stop_loop() - cancel button, exit report program
***************************************************************
*!*********************************************************************
*
```

```
*!
*!        Function: _STOP_LOOP
*!
*!       Called by: CASRPTS.PRG
*!
*!*************************************************************************
*
FUNCTION _stop_loop
#REGION 1

dont_stop = .F.

RETURN

***************** end of procedures/functions *******************
*: EOF: CASRPTS.PRG
```

APPENDIX M

PROGRAM TO INFORM THE USER OF PROGRAM/APPLICATION ERRORS

1. Program: caserr.prg

2. Author: D.Butterfield, PRC Inc.

3. Date: 5 May 1994

4. Notes: Program produced to provide the user with an error message which is displayed as a pop-up window indicating the possible execution error.

5. Usage: Program is called from most of the CAS programs.

6. Files: Uses the error number and error message that is passed to it from the program that 'discovered'/created the error.

7. Problems: None noted.

8. History:

Date	Name	Ver	Modifications	By
05/05/94	caserr	1.0	original	dbb
07/21/94	caserr	1.1	no detail	dbb

```
*********************************************************************
*
* if an error occurs inform the user
*

PARAMETERS errnum, errmsg

DEFINE WINDOW err_win FROM 21,00 TO 24,79 COLOR SCHEME 7

DO CASE
        * error: file in use by another
        CASE errnum = 108
        line1 = "File cannot be locked."
                        line2 = "Cannot append data to database."
```

```
          * error: record in use by another
          CASE errnum = 109 .OR. errnum = 130
                    line1 = "Record cannot be locked."
                    line2 = "Cannot append data to database."
          * unknown error
          OTHERWISE
                    line1 = errmsg
                    line2 = "See Your System Administrator."
ENDCASE

* activate the error window
ACTIVATE WINDOW err_win
* report an error
@ 0, (WCOLS() - LEN(line1))/2 SAY line1
@ 1, (WCOLS() - LEN(line2))/2 SAY line2

* pause
WAIT WINDOW

* release the message window
RELEASE WINDOW err_win
```

APPENDIX N

PROGRAM PROVIDING SQL QUERIES FOR THE TASK ASSESSMENT CONSISTENCY REPORT

1. This program was used in the process of validating the CAS task lists. It is not needed in the routine data processing using the AGTFS. It has been deleted from the CAS menu and cannot be directly accessed to compile a report. However, the program, itself, has not been deleted from the CAS system and database. A listing of the program may be obtained by accessing the file "cascrpt1.prg".

APPENDIX O

PROGRAM PROVIDING SQL QUERIES FOR THE TASK ASSESSMENT DISTRIBUTION REPORT

1. Program: cascrpt2.prg

2. Author: D.Butterfield, PRC Inc.

3. Date: 20 April 1994

4. Notes: Program produced to generate the necessary SQL query from the selections requested by the user in the casrpts.prg program. Provides the data for the task assessment distribution report. Each case statement is a different SQL query based on the 'All' selections.

5. Usage: Program is called from casrpts.prg. Program can not be executed by itself. However, the SQL statements may be used by cut and paste and then replacing the variables with selection criteria.

6. Files: Does not use files, but it does create two temporary CURSOR files for use. The first contains the requested data selections which are then analyzed with results placed into the second CURSOR for use by the report form casrpt2.

7. Problems: None noted.

8. History:

Date	Name	Ver	Modifications	By
04/20/94	cascrpt2	1.0	original	dbb
07/21/94	cascrpt2	1.2	many without detail	dbb
10/06/94	cascrpt2	1.3	added 'All' rotations selection	dbb

```
*************************************************************************
* task assessment distribution report
*
* different selection criteria changes report output.
* all of the case statements are near identical except for which
* variable is on and which is not. at each case statement the number
* variable is on and which is not. at each case statement the number
* indicates which of the four variables is an 'All' and which is not.
* an 'All' is indicated as a '0' and a NOT 'All' is indicated as a '1'.

IF sel_rota != "All"
        do case
                * 00000
                case moc_cs = "All" and mmission = "All" and mtrng_day = "All" and
munit_obs = "All" and mechlev = "All"
                        select a.task_no, ;
                                        a.score, ;
                                        d.task_desc ;
                                from casdata a, ;
                                        (sel_book) b, ;
                                        (sel_level) c, ;
                                        casdesc d ;
                                where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and a.task_no = d.task_no ;
                                        and a.score != 0 ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                order by a.task_no, a.score ;
                                into CURSOR distrib
                * 00001
                case moc_cs = "All" and mmission = "All" and mtrng_day = "All" and
munit_obs = "All" and mechlev != "All"
                        select a.task_no, ;
                                        a.score, ;
                                        d.task_desc ;
                                from casdata a, ;
                                        (sel_book) b, ;
                                        (sel_level) c, ;
                                        casdesc d ;
                                where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and a.task_no = d.task_no ;
                                        and a.score != 0 ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.echelon = sel_echlev ;
                                order by a.task_no, a.score ;
                                into CURSOR distrib
                * 00010
                case moc_cs = "All" and mmission = "All" and mtrng_day = "All" and
munit_obs != "All" and mechlev = "All"
                        select a.task_no, ;
                                a.score, ;
                                d.task_desc ;
                            from casdata a, ;
                                (sel_book) b, ;
                                (sel_level) c, ;
                                casdesc d ;
                            where a.task_no = b.task_no ;
                                and a.task_no = c.task_no ;
                                and a.task_no = d.task_no ;
                                and a.score != 0 ;
                                and b.tk_id_type = sel_type ;
                                and a.task_id = sel_tkid ;
                                and a.rotation = sel_rota ;
                                and a.unit_obs = munit_obs ;
                            order by a.task_no, a.score ;
                            into CURSOR distrib
                * 00011
                case moc_cs = "All" and mmission = "All" and mtrng_day = "All" and
munit_obs != "All" and mechlev != "All"
                        select a.task_no, ;
                                a.score, ;
                                d.task_desc ;
                            from casdata a, ;
                                (sel_book) b, ;
                                (sel_level) c, ;
                                casdesc d ;
                            where a.task_no = b.task_no ;
                                and a.task_no = c.task_no ;
                                and a.task_no = d.task_no ;
                                and a.score != 0 ;
                                and b.tk_id_type = sel_type ;
                                and a.task_id = sel_tkid ;
                                and a.rotation = sel_rota ;
                                and a.unit_obs = munit_obs ;
                                and a.echelon = sel_echlev ;
                            order by a.task_no, a.score ;
                            into CURSOR distrib
                * 00100
                case moc_cs = "All" and mmission = "All" and mtrng_day != "All" and
munit_obs = "All" and mechlev = "All"
                        select a.task_no, ;
```

```
                                        a.score, ;
                                        d.task_desc ;
                                  from casdata a, ;
                                        (sel_book) b, ;
                                        (sel_level) c, ;
                                        casdesc d ;
                                  where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and a.task_no = d.task_no ;
                                        and a.score != 0 ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.trng_day = mtrng_day ;
                                  order by a.task_no, a.score ;
                                  into CURSOR distrib
              * 00101
              case moc_cs = "All" and mmission = "All" and mtrng_day != "All" and
munit_obs = "All" and mechlev != "All"
                            select a.task_no, ;
                                        a.score, ;
                                        d.task_desc ;
                                  from casdata a, ;
                                        (sel_book) b, ;
                                        (sel_level) c, ;
                                        casdesc d ;
                                  where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and a.task_no = d.task_no ;
                                        and a.score != 0 ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.trng_day = mtrng_day ;
                                        and a.echelon = sel_echlev ;
                                  order by a.task_no, a.score ;
                                  into CURSOR distrib
              * 00110
              case moc_cs = "All" and mmission = "All" and mtrng_day != "All" and
munit_obs != "All" and mechlev = "All"
                            select a.task_no, ;
                                        a.score, ;
                                        d.task_desc ;
                                  from casdata a, ;
                                        (sel_book) b, ;
                                        (sel_level) c, ;
                                        casdesc d ;
                              where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and a.task_no = d.task_no ;
                                        and a.score != 0 ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.trng_day = mtrng_day ;
                                        and a.unit_obs = munit_obs ;
                              order by a.task_no, a.score ;
                              into CURSOR distrib
              * 00111
              case moc_cs = "All" and mmission = "All" and mtrng_day != "All" and
munit_obs != "All" and mechlev != "All"
                         select a.task_no, ;
                                        a.score, ;
                                        d.task_desc ;
                              from casdata a, ;
                                        (sel_book) b, ;
                                        (sel_level) c, ;
                                        casdesc d ;
                              where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and a.task_no = d.task_no ;
                                        and a.score != 0 ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.trng_day = mtrng_day ;
                                        and a.unit_obs = munit_obs ;
                                        and a.echelon = sel_echlev ;
                              order by a.task_no, a.score ;
                              into CURSOR distrib
              * 01000
              case moc_cs = "All" and mmission != "All" and mtrng_day = "All" and
munit_obs = "All" and mechlev = "All"
                         select a.task_no, ;
                                        a.score, ;
                                        d.task_desc ;
                              from casdata a, ;
                                        (sel_book) b, ;
                                        (sel_level) c, ;
                                        casdesc d ;
                                        where a.task_no = b.task_no ;
                                                and a.task_no = c.task_no ;
                                                and a.task_no = d.task_no ;
                                                and a.score != 0 ;
                                                and b.tk_id_type = sel_type ;
                                                and a.task_id = sel_tkid ;
                                                and a.rotation = sel_rota ;
                                                and a.mission = mmission ;
                                        order by a.task_no, a.score ;
                                        into CURSOR distrib
                        * 01001
                        case moc_cs = "All" and mmission != "All" and mtrng_day = "All" and
munit_obs = "All" and mechlev != "All"
                                select a.task_no, ;
                                                a.score, ;
                                                d.task_desc ;
                                        from casdata a, ;
                                                (sel_book) b, ;
                                                (sel_level) c, ;
                                                casdesc d ;
                                        where a.task_no = b.task_no ;
                                                and a.task_no = c.task_no ;
                                                and a.task_no = d.task_no ;
                                                and a.score != 0 ;
                                                and b.tk_id_type = sel_type ;
                                                and a.task_id = sel_tkid ;
                                                and a.rotation = sel_rota ;
                                                and a.mission = mmission ;
                                                and a.echelon = sel_echlev ;
                                        order by a.task_no, a.score ;
                                        into CURSOR distrib
                        * 01010
                        case moc_cs = "All" and mmission != "All" and mtrng_day = "All" and
munit_obs != "All" and mechlev = "All"
                                select a.task_no, ;
                                                a.score, ;
                                                d.task_desc ;
                                        from casdata a, ;
                                                (sel_book) b, ;
                                                (sel_level) c, ;
                                                casdesc d ;
                                        where a.task_no = b.task_no ;
                                                and a.task_no = c.task_no ;
                                                and a.task_no = d.task_no ;
                                                and a.score != 0 ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.mission = mmission ;
                                        and a.unit_obs = munit_obs ;
                              order by a.task_no, a.score ;
                              into CURSOR distrib
            * 01011
            case moc_cs = "All" and mmission != "All" and mtrng_day = "All" and
munit_obs != "All" and mechlev != "All"
                        select a.task_no, ;
                                        a.score, ;
                                        d.task_desc ;
                              from casdata a, ;
                                        (sel_book) b, ;
                                        (sel_level) c, ;
                                        casdesc d ;
                              where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and a.task_no = d.task_no ;
                                        and a.score != 0 ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.mission = mmission ;
                                        and a.unit_obs = munit_obs ;
                                        and a.echelon = sel_echlev ;
                              order by a.task_no, a.score ;
                              into CURSOR distrib
            * 01100
            case moc_cs = "All" and mmission != "All" and mtrng_day != "All" and
munit_obs = "All" and mechlev = "All"
                        select a.task_no, ;
                                        a.score, ;
                                        d.task_desc ;
                              from casdata a, ;
                                        (sel_book) b, ;
                                        (sel_level) c, ;
                                        casdesc d ;
                              where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and a.task_no = d.task_no ;
                                        and a.score != 0 ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.mission = mmission ;
                                        and a.trng_day = mtrng_day ;
                                order by a.task_no, a.score ;
                                into CURSOR distrib
                * 01101
                case moc_cs = "All" and mmission != "All" and mtrng_day != "All" and
munit_obs = "All" and mechlev != "All"
                        select a.task_no, ;
                                        a.score, ;
                                        d.task_desc ;
                                from casdata a, ;
                                        (sel_book) b, ;
                                        (sel_level) c, ;
                                        casdesc d ;
                                where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and a.task_no = d.task_no ;
                                        and a.score != 0 ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.mission = mmission ;
                                        and a.trng_day = mtrng_day ;
                                        and a.echelon = sel_echlev ;
                                order by a.task_no, a.score ;
                                into CURSOR distrib
                * 01110
                case moc_cs = "All" and mmission != "All" and mtrng_day != "All" and
munit_obs != "All" and mechlev = "All"
                        select a.task_no, ;
                                        a.score, ;
                                        d.task_desc ;
                                from casdata a, ;
                                        (sel_book) b, ;
                                        (sel_level) c, ;
                                        casdesc d ;
                                where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and a.task_no = d.task_no ;
                                        and a.score != 0 ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.mission = mmission ;
                                        and a.trng_day = mtrng_day ;
                                        and a.unit_obs = munit_obs ;
                                 order by a.task_no, a.score ;
                                 into CURSOR distrib
                  * 01111
                  case moc_cs = "All" and mmission != "All" and mtrng_day != "All" and
munit_obs != "All" and mechlev != "All"
                           select a.task_no, ;
                                        a.score, ;
                                        d.task_desc ;
                                 from casdata a, ;
                                        (sel_book) b, ;
                                        (sel_level) c, ;
                                        casdesc d ;
                                 where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and a.task_no = d.task_no ;
                                        and a.score != 0 ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.mission = mmission ;
                                        and a.trng_day = mtrng_day ;
                                        and a.unit_obs = munit_obs ;
                                        and a.echelon = sel_echlev ;
                                 order by a.task_no, a.score ;
                                 into CURSOR distrib
                  * 10000
                  case moc_cs != "All" and mmission = "All" and mtrng_day = "All" and
munit_obs = "All" and mechlev = "All"
                           select a.task_no, ;
                                        a.score, ;
                                        d.task_desc ;
                                 from casdata a, ;
                                        (sel_book) b, ;
                                        (sel_level) c, ;
                                        casdesc d ;
                                 where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and a.task_no = d.task_no ;
                                        and a.score != 0 ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.oc_cs = moc_cs ;
                                        order by a.task_no, a.score ;
                                        into CURSOR distrib
                        * 10001
                        case moc_cs != "All" and mmission = "All" and mtrng_day = "All" and
munit_obs = "All" and mechlev != "All"
                                select a.task_no, ;
                                                a.score, ;
                                                d.task_desc ;
                                        from casdata a, ;
                                                (sel_book) b, ;
                                                (sel_level) c, ;
                                                casdesc d ;
                                        where a.task_no = b.task_no ;
                                                and a.task_no = c.task_no ;
                                                and a.task_no = d.task_no ;
                                                and a.score != 0 ;
                                                and b.tk_id_type = sel_type ;
                                                and a.task_id = sel_tkid ;
                                                and a.rotation = sel_rota ;
                                                and a.oc_cs = moc_cs ;
                                                and a.echelon = sel_echlev ;
                                        order by a.task_no, a.score ;
                                        into CURSOR distrib
                        * 10100
                        case moc_cs != "All" and mmission = "All" and mtrng_day != "All" and
munit_obs = "All" and mechlev = "All"
                                select a.task_no, ;
                                                a.score, ;
                                                d.task_desc ;
                                        from casdata a, ;
                                                (sel_book) b, ;
                                                (sel_level) c, ;
                                                casdesc d ;
                                        where a.task_no = b.task_no ;
                                                and a.task_no = c.task_no ;
                                                and a.task_no = d.task_no ;
                                                and a.score != 0 ;
                                                and b.tk_id_type = sel_type ;
                                                and a.task_id = sel_tkid ;
                                                and a.rotation = sel_rota ;
                                                and a.oc_cs = moc_cs ;
                                                and a.trng_day = mtrng_day ;
                                        order by a.task_no, a.score ;
                                        into CURSOR distrib
                        * 10101
```

```
                    case moc_cs != "All" and mmission = "All" and mtrng_day != "All" and
munit_obs = "All" and mechlev != "All"
                              select a.task_no, ;
                                          a.score, ;
                                          d.task_desc ;
                                    from casdata a, ;
                                          (sel_book) b, ;
                                          (sel_level) c, ;
                                          casdesc d ;
                                    where a.task_no = b.task_no ;
                                          and a.task_no = c.task_no ;
                                          and a.task_no = d.task_no ;
                                          and a.score != 0 ;
                                          and b.tk_id_type = sel_type ;
                                          and a.task_id = sel_tkid ;
                                          and a.rotation = sel_rota ;
                                          and a.oc_cs = moc_cs ;
                                          and a.trng_day = mtrng_day ;
                                          and a.echelon = sel_echlev ;
                                    order by a.task_no, a.score ;
                                    into CURSOR distrib
                    * 10110
                    case moc_cs != "All" and mmission = "All" and mtrng_day != "All" and
munit_obs != "All" and mechlev = "All"
                              select a.task_no, ;
                                          a.score, ;
                                          d.task_desc ;
                                    from casdata a, ;
                                          (sel_book) b, ;
                                          (sel_level) c, ;
                                          casdesc d ;
                                    where a.task_no = b.task_no ;
                                          and a.task_no = c.task_no ;
                                          and a.task_no = d.task_no ;
                                          and a.score != 0 ;
                                          and b.tk_id_type = sel_type ;
                                          and a.task_id = sel_tkid ;
                                          and a.rotation = sel_rota ;
                                          and a.oc_cs = moc_cs ;
                                          and a.trng_day = mtrng_day ;
                                          and a.unit_obs = munit_obs ;
                                    order by a.task_no, a.score ;
                                    into CURSOR distrib
                    * 10111
                    case moc_cs != "All" and mmission = "All" and mtrng_day != "All" and
```

```
munit_obs != "All" and mechlev != "All"
                                        select a.task_no, ;
                                                            a.score, ;
                                                            d.task_desc ;
                                                  from casdata a, ;
                                                            (sel_book) b, ;
                                                            (sel_level) c, ;
                                                            casdesc d ;
                                                  where a.task_no = b.task_no ;
                                                            and a.task_no = c.task_no ;
                                                            and a.task_no = d.task_no ;
                                                            and a.score != 0 ;
                                                            and b.tk_id_type = sel_type ;
                                                            and a.task_id = sel_tkid ;
                                                            and a.rotation = sel_rota ;
                                                            and a.oc_cs = moc_cs ;
                                                            and a.trng_day = mtrng_day ;
                                                            and a.unit_obs = munit_obs ;
                                                            and a.echelon = sel_echlev ;
                                                  order by a.task_no, a.score ;
                                                  into CURSOR distrib
                              * 11000
                              case moc_cs != "All" and mmission != "All" and mtrng_day = "All" and
munit_obs = "All" and mechlev = "All"
                                        select a.task_no, ;
                                                            a.score, ;
                                                            d.task_desc ;
                                                  from casdata a, ;
                                                            (sel_book) b, ;
                                                            (sel_level) c, ;
                                                            casdesc d ;
                                                  where a.task_no = b.task_no ;
                                                            and a.task_no = c.task_no ;
                                                            and a.task_no = d.task_no ;
                                                            and a.score != 0 ;
                                                            and b.tk_id_type = sel_type ;
                                                            and a.task_id = sel_tkid ;
                                                            and a.rotation = sel_rota ;
                                                            and a.oc_cs = moc_cs ;
                                                            and a.mission = mmission ;
                                                  order by a.task_no, a.score ;
                                                  into CURSOR distrib
                              * 11001
                              case moc_cs != "All" and mmission != "All" and mtrng_day = "All" and
munit_obs = "All" and mechlev != "All"
```

```
                    select a.task_no, ;
                                a.score, ;
                                d.task_desc ;
                          from casdata a, ;
                                (sel_book) b, ;
                                (sel_level) c, ;
                                casdesc d ;
                          where a.task_no = b.task_no ;
                                and a.task_no = c.task_no ;
                                and a.task_no = d.task_no ;
                                and a.score != 0 ;
                                and b.tk_id_type = sel_type ;
                                and a.task_id = sel_tkid ;
                                and a.rotation = sel_rota ;
                                and a.oc_cs = moc_cs ;
                                and a.mission = mmission ;
                                and a.echelon = sel_echlev ;
                          order by a.task_no, a.score ;
                          into CURSOR distrib
              * 11010
              case moc_cs != "All" and mmission != "All" and mtrng_day = "All" and
munit_obs != "All" and mechlev = "All"
                    select a.task_no, ;
                                a.score, ;
                                d.task_desc ;
                          from casdata a, ;
                                (sel_book) b, ;
                                (sel_level) c, ;
                                casdesc d ;
                          where a.task_no = b.task_no ;
                                and a.task_no = c.task_no ;
                                and a.task_no = d.task_no ;
                                and a.score != 0 ;
                                and b.tk_id_type = sel_type ;
                                and a.task_id = sel_tkid ;
                                and a.rotation = sel_rota ;
                                and a.oc_cs = moc_cs ;
                                and a.mission = mmission ;
                                and a.unit_obs = munit_obs ;
                          order by a.task_no, a.score ;
                          into CURSOR distrib
              * 11011
              case moc_cs != "All" and mmission != "All" and mtrng_day = "All" and
munit_obs != "All" and mechlev != "All"
                    select a.task_no, ;
```

```
                                    a.score, ;
                                    d.task_desc ;
                              from casdata a, ;
                                    (sel_book) b, ;
                                    (sel_level) c, ;
                                    casdesc d ;
                              where a.task_no = b.task_no ;
                                    and a.task_no = c.task_no ;
                                    and a.task_no = d.task_no ;
                                    and a.score != 0 ;
                                    and b.tk_id_type = sel_type ;
                                    and a.task_id = sel_tkid ;
                                    and a.rotation = sel_rota ;
                                    and a.oc_cs = moc_cs ;
                                    and a.mission = mmission ;
                                    and a.unit_obs = munit_obs ;
                                    and a.echelon = sel_echlev ;
                              order by a.task_no, a.score ;
                              into CURSOR distrib
                  * 11100
                  case moc_cs != "All" and mmission != "All" and mtrng_day != "All" and
munit_obs = "All" and mechlev = "All"
                        select a.task_no, ;
                                    a.score, ;
                                    d.task_desc ;
                              from casdata a, ;
                                    (sel_book) b, ;
                                    (sel_level) c, ;
                                    casdesc d ;
                              where a.task_no = b.task_no ;
                                    and a.task_no = c.task_no ;
                                    and a.task_no = d.task_no ;
                                    and a.score != 0 ;
                                    and b.tk_id_type = sel_type ;
                                    and a.task_id = sel_tkid ;
                                    and a.rotation = sel_rota ;
                                    and a.oc_cs = moc_cs ;
                                    and a.mission = mmission ;
                                    and a.trng_day = trng_day ;
                              order by a.task_no, a.score ;
                              into CURSOR distrib
                  * 11101
                  case moc_cs != "All" and mmission != "All" and mtrng_day != "All" and
munit_obs = "All" and mechlev != "All"
                        select a.task_no, ;
```

```
                                        a.score, ;
                                        d.task_desc ;
                              from casdata a, ;
                                        (sel_book) b, ;
                                        (sel_level) c, ;
                                        casdesc d ;
                              where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and a.task_no = d.task_no ;
                                        and a.score != 0 ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.oc_cs = moc_cs ;
                                        and a.mission = mmission ;
                                        and a.trng_day = trng_day ;
                                        and a.echelon = sel_echlev ;
                              order by a.task_no, a.score ;
                              into CURSOR distrib
              * 11110
              case moc_cs != "All" and mmission != "All" and mtrng_day != "All" and
munit_obs != "All" and mechlev = "All"
                        select a.task_no, ;
                                        a.score, ;
                                        d.task_desc ;
                              from casdata a, ;
                                        (sel_book) b, ;
                                        (sel_level) c, ;
                                        casdesc d ;
                              where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and a.task_no = d.task_no ;
                                        and a.score != 0 ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.oc_cs = moc_cs ;
                                        and a.mission = mmission ;
                                        and a.trng_day = trng_day ;
                                        and a.unit_obs = munit_obs ;
                              order by a.task_no, a.score ;
                              into CURSOR distrib
              * 11111
              case moc_cs != "All" and mmission != "All" and mtrng_day != "All" and
munit_obs != "All" and mechlev != "All"
```

```
                              select a.task_no, ;
                                             a.score, ;
                                             d.task_desc ;
                                     from casdata a, ;
                                             (sel_book) b, ;
                                             (sel_level) c, ;
                                             casdesc d ;
                                     where a.task_no = b.task_no ;
                                             and a.task_no = c.task_no ;
                                             and a.task_no = d.task_no ;
                                             and a.score != 0 ;
                                             and b.tk_id_type = sel_type ;
                                             and a.task_id = sel_tkid ;
                                             and a.rotation = sel_rota ;
                                             and a.oc_cs = moc_cs ;
                                             and a.mission = mmission ;
                                             and a.trng_day = trng_day ;
                                             and a.unit_obs = munit_obs ;
                                             and a.echelon = sel_echlev ;
                                     order by a.task_no, a.score ;
                                     into CURSOR distrib
       endcase
ELSE
* select all rotations
       do case
              * 00000
              case moc_cs = "All" and mmission = "All" and mtrng_day = "All" and
munit_obs = "All" and mechlev = "All"
                              select a.task_no, ;
                                             a.score, ;
                                             d.task_desc ;
                                     from casdata a, ;
                                             (sel_book) b, ;
                                             (sel_level) c, ;
                                             casdesc d ;
                                     where a.task_no = b.task_no ;
                                             and a.task_no = c.task_no ;
                                             and a.task_no = d.task_no ;
                                             and a.score != 0 ;
                                             and b.tk_id_type = sel_type ;
                                             and a.task_id = sel_tkid ;
                                     order by a.task_no, a.score ;
                                     into CURSOR distrib
              * 00001
              case moc_cs = "All" and mmission = "All" and mtrng_day = "All" and
```

```
munit_obs = "All" and mechlev != "All"
                              select a.task_no, ;
                                             a.score, ;
                                             d.task_desc ;
                                     from casdata a, ;
                                             (sel_book) b, ;
                                             (sel_level) c, ;
                                             casdesc d ;
                                     where a.task_no = b.task_no ;
                                             and a.task_no = c.task_no ;
                                             and a.task_no = d.task_no ;
                                             and a.score != 0 ;
                                             and b.tk_id_type = sel_type ;
                                             and a.task_id = sel_tkid ;
                                             and a.echelon = sel_echlev ;
                                     order by a.task_no, a.score ;
                                     into CURSOR distrib
              * 00010
              case moc_cs = "All" and mmission = "All" and mtrng_day = "All" and
munit_obs != "All" and mechlev = "All"
                              select a.task_no, ;
                                             a.score, ;
                                             d.task_desc ;
                                     from casdata a, ;
                                             (sel_book) b, ;
                                             (sel_level) c, ;
                                             casdesc d ;
                                     where a.task_no = b.task_no ;
                                             and a.task_no = c.task_no ;
                                             and a.task_no = d.task_no ;
                                             and a.score != 0 ;
                                             and b.tk_id_type = sel_type ;
                                             and a.task_id = sel_tkid ;
                                             and a.unit_obs = munit_obs ;
                                     order by a.task_no, a.score ;
                                     into CURSOR distrib
              * 00011
              case moc_cs = "All" and mmission = "All" and mtrng_day = "All" and
munit_obs != "All" and mechlev != "All"
                              select a.task_no, ;
                                             a.score, ;
                                             d.task_desc ;
                                     from casdata a, ;
                                             (sel_book) b, ;
                                             (sel_level) c, ;
```

```
                                        casdesc d ;
                              where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and a.task_no = d.task_no ;
                                        and a.score != 0 ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.unit_obs = munit_obs ;
                                        and a.echelon = sel_echlev ;
                              order by a.task_no, a.score ;
                              into CURSOR distrib
            * 00100
            case moc_cs = "All" and mmission = "All" and mtrng_day != "All" and
munit_obs = "All" and mechlev = "All"
                    select a.task_no, ;
                                        a.score, ;
                                        d.task_desc ;
                              from casdata a, ;
                                        (sel_book) b, ;
                                        (sel_level) c, ;
                                        casdesc d ;
                              where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and a.task_no = d.task_no ;
                                        and a.score != 0 ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.trng_day = mtrng_day ;
                              order by a.task_no, a.score ;
                              into CURSOR distrib
            * 00101
            case moc_cs = "All" and mmission = "All" and mtrng_day != "All" and
munit_obs = "All" and mechlev != "All"
                    select a.task_no, ;
                                        a.score, ;
                                        d.task_desc ;
                              from casdata a, ;
                                        (sel_book) b, ;
                                        (sel_level) c, ;
                                        casdesc d ;
                              where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and a.task_no = d.task_no ;
                                        and a.score != 0 ;
                                        and b.tk_id_type = sel_type ;
```

```
                                                  and a.task_id = sel_tkid ;
                                                  and a.trng_day = mtrng_day ;
                                                  and a.echelon = sel_echlev ;
                                          order by a.task_no, a.score ;
                                          into CURSOR distrib
                          * 00110
                          case moc_cs = "All" and mmission = "All" and mtrng_day != "All" and
munit_obs != "All" and mechlev = "All"
                                  select a.task_no, ;
                                                  a.score, ;
                                                  d.task_desc ;
                                          from casdata a, ;
                                                  (sel_book) b, ;
                                                  (sel_level) c, ;
                                                  casdesc d ;
                                          where a.task_no = b.task_no ;
                                                  and a.task_no = c.task_no ;
                                                  and a.task_no = d.task_no ;
                                                  and a.score != 0 ;
                                                  and b.tk_id_type = sel_type ;
                                                  and a.task_id = sel_tkid ;
                                                  and a.trng_day = mtrng_day ;
                                                  and a.unit_obs = munit_obs ;
                                          order by a.task_no, a.score ;
                                          into CURSOR distrib
                          * 00111
                          case moc_cs = "All" and mmission = "All" and mtrng_day != "All" and
munit_obs != "All" and mechlev != "All"
                                  select a.task_no, ;
                                                  a.score, ;
                                                  d.task_desc ;
                                          from casdata a, ;
                                                  (sel_book) b, ;
                                                  (sel_level) c, ;
                                                  casdesc d ;
                                          where a.task_no = b.task_no ;
                                                  and a.task_no = c.task_no ;
                                                  and a.task_no = d.task_no ;
                                                  and a.score != 0 ;
                                                  and b.tk_id_type = sel_type ;
                                                  and a.task_id = sel_tkid ;
                                                  and a.trng_day = mtrng_day ;
                                                  and a.unit_obs = munit_obs ;
                                                  and a.echelon = sel_echlev ;
                                          order by a.task_no, a.score ;
```

```
                                        into CURSOR distrib
                        * 01000
                        case moc_cs = "All" and mmission != "All" and mtrng_day = "All" and
munit_obs = "All" and mechlev = "All"
                                select a.task_no, ;
                                                a.score, ;
                                                d.task_desc ;
                                        from casdata a, ;
                                                (sel_book) b, ;
                                                (sel_level) c, ;
                                                casdesc d ;
                                        where a.task_no = b.task_no ;
                                                and a.task_no = c.task_no ;
                                                and a.task_no = d.task_no ;
                                                and a.score != 0 ;
                                                and b.tk_id_type = sel_type ;
                                                and a.task_id = sel_tkid ;
                                                and a.mission = mmission ;
                                        order by a.task_no, a.score ;
                                        into CURSOR distrib
                        * 01001
                        case moc_cs = "All" and mmission != "All" and mtrng_day = "All" and
munit_obs = "All" and mechlev != "All"
                                select a.task_no, ;
                                                a.score, ;
                                                d.task_desc ;
                                        from casdata a, ;
                                                (sel_book) b, ;
                                                (sel_level) c, ;
                                                casdesc d ;
                                        where a.task_no = b.task_no ;
                                                and a.task_no = c.task_no ;
                                                and a.task_no = d.task_no ;
                                                and a.score != 0 ;
                                                and b.tk_id_type = sel_type ;
                                                and a.task_id = sel_tkid ;
                                                and a.mission = mmission ;
                                                and a.echelon = sel_echlev ;
                                        order by a.task_no, a.score ;
                                        into CURSOR distrib
                        * 01010
                        case moc_cs = "All" and mmission != "All" and mtrng_day = "All" and
munit_obs != "All" and mechlev = "All"
                                select a.task_no, ;
                                                a.score, ;
                                        d.task_desc ;
                              from casdata a, ;
                                        (sel_book) b, ;
                                        (sel_level) c, ;
                                        casdesc d ;
                              where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and a.task_no = d.task_no ;
                                        and a.score != 0 ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.mission = mmission ;
                                        and a.unit_obs = munit_obs ;
                              order by a.task_no, a.score ;
                              into CURSOR distrib
                * 01011
                case moc_cs = "All" and mmission != "All" and mtrng_day = "All" and
munit_obs != "All" and mechlev != "All"
                         select a.task_no, ;
                                        a.score, ;
                                        d.task_desc ;
                              from casdata a, ;
                                        (sel_book) b, ;
                                        (sel_level) c, ;
                                        casdesc d ;
                              where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and a.task_no = d.task_no ;
                                        and a.score != 0 ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.mission = mmission ;
                                        and a.unit_obs = munit_obs ;
                                        and a.echelon = sel_echlev ;
                              order by a.task_no, a.score ;
                              into CURSOR distrib
                * 01100
                case moc_cs = "All" and mmission != "All" and mtrng_day != "All" and
munit_obs = "All" and mechlev = "All"
                         select a.task_no, ;
                                        a.score, ;
                                        d.task_desc ;
                              from casdata a, ;
                                        (sel_book) b, ;
                                        (sel_level) c, ;
                                        casdesc d ;
                                  where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and a.task_no = d.task_no ;
                                        and a.score != 0 ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.mission = mmission ;
                                        and a.trng_day = mtrng_day ;
                                  order by a.task_no, a.score ;
                                  into CURSOR distrib
                    * 01101
                    case moc_cs = "All" and mmission != "All" and mtrng_day != "All" and
munit_obs = "All" and mechlev != "All"
                            select a.task_no, ;
                                        a.score, ;
                                        d.task_desc ;
                                  from casdata a, ;
                                        (sel_book) b, ;
                                        (sel_level) c, ;
                                        casdesc d ;
                                  where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and a.task_no = d.task_no ;
                                        and a.score != 0 ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.mission = mmission ;
                                        and a.trng_day = mtrng_day ;
                                        and a.echelon = sel_echlev ;
                                  order by a.task_no, a.score ;
                                  into CURSOR distrib
                    * 01110
                    case moc_cs = "All" and mmission != "All" and mtrng_day != "All" and
munit_obs != "All" and mechlev = "All"
                            select a.task_no, ;
                                        a.score, ;
                                        d.task_desc ;
                                  from casdata a, ;
                                        (sel_book) b, ;
                                        (sel_level) c, ;
                                        casdesc d ;
                                  where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and a.task_no = d.task_no ;
                                        and a.score != 0 ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.mission = mmission ;
                                        and a.trng_day = mtrng_day ;
                                        and a.unit_obs = munit_obs ;
                                order by a.task_no, a.score ;
                                into CURSOR distrib
                * 01111
                case moc_cs = "All" and mmission != "All" and mtrng_day != "All" and
munit_obs != "All" and mechlev != "All"
                        select a.task_no, ;
                                        a.score, ;
                                        d.task_desc ;
                                from casdata a, ;
                                        (sel_book) b, ;
                                        (sel_level) c, ;
                                        casdesc d ;
                                where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and a.task_no = d.task_no ;
                                        and a.score != 0 ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.mission = mmission ;
                                        and a.trng_day = mtrng_day ;
                                        and a.unit_obs = munit_obs ;
                                        and a.echelon = sel_echlev ;
                                order by a.task_no, a.score ;
                                into CURSOR distrib
                * 10000
                case moc_cs != "All" and mmission = "All" and mtrng_day = "All" and
munit_obs = "All" and mechlev = "All"
                        select a.task_no, ;
                                        a.score, ;
                                        d.task_desc ;
                                from casdata a, ;
                                        (sel_book) b, ;
                                        (sel_level) c, ;
                                        casdesc d ;
                                where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and a.task_no = d.task_no ;
                                        and a.score != 0 ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.oc_cs = moc_cs ;
                              order by a.task_no, a.score ;
                              into CURSOR distrib
          * 10001
          case moc_cs != "All" and mmission = "All" and mtrng_day = "All" and
munit_obs = "All" and mechlev != "All"
                    select a.task_no, ;
                                        a.score, ;
                                        d.task_desc ;
                              from casdata a, ;
                                        (sel_book) b, ;
                                        (sel_level) c, ;
                                        casdesc d ;
                              where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and a.task_no = d.task_no ;
                                        and a.score != 0 ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.oc_cs = moc_cs ;
                                        and a.echelon = sel_echlev ;
                              order by a.task_no, a.score ;
                              into CURSOR distrib
          * 10100
          case moc_cs != "All" and mmission = "All" and mtrng_day != "All" and
munit_obs = "All" and mechlev = "All"
                    select a.task_no, ;
                                        a.score, ;
                                        d.task_desc ;
                              from casdata a, ;
                                        (sel_book) b, ;
                                        (sel_level) c, ;
                                        casdesc d ;
                              where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and a.task_no = d.task_no ;
                                        and a.score != 0 ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.oc_cs = moc_cs ;
                                        and a.trng_day = mtrng_day ;
                              order by a.task_no, a.score ;
                              into CURSOR distrib
          * 10101
```

```
                case moc_cs != "All" and mmission = "All" and mtrng_day != "All" and
munit_obs = "All" and mechlev != "All"
                        select a.task_no, ;
                                    a.score, ;
                                    d.task_desc ;
                              from casdata a, ;
                                    (sel_book) b, ;
                                    (sel_level) c, ;
                                    casdesc d ;
                              where a.task_no = b.task_no ;
                                    and a.task_no = c.task_no ;
                                    and a.task_no = d.task_no ;
                                    and a.score != 0 ;
                                    and b.tk_id_type = sel_type ;
                                    and a.task_id = sel_tkid ;
                                    and a.oc_cs = moc_cs ;
                                    and a.trng_day = mtrng_day ;
                                    and a.echelon = sel_echlev ;
                              order by a.task_no, a.score ;
                              into CURSOR distrib
                * 10110
                case moc_cs != "All" and mmission = "All" and mtrng_day != "All" and
munit_obs != "All" and mechlev = "All"
                        select a.task_no, ;
                                    a.score, ;
                                    d.task_desc ;
                              from casdata a, ;
                                    (sel_book) b, ;
                                    (sel_level) c, ;
                                    casdesc d ;
                              where a.task_no = b.task_no ;
                                    and a.task_no = c.task_no ;
                                    and a.task_no = d.task_no ;
                                    and a.score != 0 ;
                                    and b.tk_id_type = sel_type ;
                                    and a.task_id = sel_tkid ;
                                    and a.oc_cs = moc_cs ;
                                    and a.trng_day = mtrng_day ;
                                    and a.unit_obs = munit_obs ;
                              order by a.task_no, a.score ;
                              into CURSOR distrib
                * 10111
                case moc_cs != "All" and mmission = "All" and mtrng_day != "All" and
munit_obs != "All" and mechlev != "All"
                        select a.task_no, ;
```

```
                                    a.score, ;
                                    d.task_desc ;
                              from casdata a, ;
                                    (sel_book) b, ;
                                    (sel_level) c, ;
                                    casdesc d ;
                              where a.task_no = b.task_no ;
                                    and a.task_no = c.task_no ;
                                    and a.task_no = d.task_no ;
                                    and a.score != 0 ;
                                    and b.tk_id_type = sel_type ;
                                    and a.task_id = sel_tkid ;
                                    and a.oc_cs = moc_cs ;
                                    and a.trng_day = mtrng_day ;
                                    and a.unit_obs = munit_obs ;
                                    and a.echelon = sel_echlev ;
                              order by a.task_no, a.score ;
                              into CURSOR distrib
            * 11000
            case moc_cs != "All" and mmission != "All" and mtrng_day = "All" and
munit_obs = "All" and mechlev = "All"
                        select a.task_no, ;
                                    a.score, ;
                                    d.task_desc ;
                              from casdata a, ;
                                    (sel_book) b, ;
                                    (sel_level) c, ;
                                    casdesc d ;
                              where a.task_no = b.task_no ;
                                    and a.task_no = c.task_no ;
                                    and a.task_no = d.task_no ;
                                    and a.score != 0 ;
                                    and b.tk_id_type = sel_type ;
                                    and a.task_id = sel_tkid ;
                                    and a.oc_cs = moc_cs ;
                                    and a.mission = mmission ;
                              order by a.task_no, a.score ;
                              into CURSOR distrib
            * 11001
            case moc_cs != "All" and mmission != "All" and mtrng_day = "All" and
munit_obs = "All" and mechlev != "All"
                        select a.task_no, ;
                                    a.score, ;
                                    d.task_desc ;
                              from casdata a, ;
```

```
                                        (sel_book) b, ;
                                        (sel_level) c, ;
                                        casdesc d ;
                                where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and a.task_no = d.task_no ;
                                        and a.score != 0 ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.oc_cs = moc_cs ;
                                        and a.mission = mmission ;
                                        and a.echelon = sel_echlev ;
                                order by a.task_no, a.score ;
                                into CURSOR distrib
                * 11010
                case moc_cs != "All" and mmission != "All" and mtrng_day = "All" and
munit_obs != "All" and mechlev = "All"
                        select a.task_no, ;
                                        a.score, ;
                                        d.task_desc ;
                                from casdata a, ;
                                        (sel_book) b, ;
                                        (sel_level) c, ;
                                        casdesc d ;
                                where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and a.task_no = d.task_no ;
                                        and a.score != 0 ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.oc_cs = moc_cs ;
                                        and a.mission = mmission ;
                                        and a.unit_obs = munit_obs ;
                                order by a.task_no, a.score ;
                                into CURSOR distrib
                * 11011
                case moc_cs != "All" and mmission != "All" and mtrng_day = "All" and
munit_obs != "All" and mechlev != "All"
                        select a.task_no, ;
                                        a.score, ;
                                        d.task_desc ;
                                from casdata a, ;
                                        (sel_book) b, ;
                                        (sel_level) c, ;
                                        casdesc d ;
                                        where a.task_no = b.task_no ;
                                                and a.task_no = c.task_no ;
                                                and a.task_no = d.task_no ;
                                                and a.score != 0 ;
                                                and b.tk_id_type = sel_type ;
                                                and a.task_id = sel_tkid ;
                                                and a.oc_cs = moc_cs ;
                                                and a.mission = mmission ;
                                                and a.unit_obs = munit_obs ;
                                                and a.echelon = sel_echlev ;
                                        order by a.task_no, a.score ;
                                        into CURSOR distrib
                        * 11100
                        case moc_cs != "All" and mmission != "All" and mtrng_day != "All" and
munit_obs = "All" and mechlev = "All"
                                select a.task_no, ;
                                                a.score, ;
                                                d.task_desc ;
                                        from casdata a, ;
                                                (sel_book) b, ;
                                                (sel_level) c, ;
                                                casdesc d ;
                                        where a.task_no = b.task_no ;
                                                and a.task_no = c.task_no ;
                                                and a.task_no = d.task_no ;
                                                and a.score != 0 ;
                                                and b.tk_id_type = sel_type ;
                                                and a.task_id = sel_tkid ;
                                                and a.oc_cs = moc_cs ;
                                                and a.mission = mmission ;
                                                and a.trng_day = trng_day ;
                                        order by a.task_no, a.score ;
                                        into CURSOR distrib
                        * 11101
                        case moc_cs != "All" and mmission != "All" and mtrng_day != "All" and
munit_obs = "All" and mechlev != "All"
                                select a.task_no, ;
                                                a.score, ;
                                                d.task_desc ;
                                        from casdata a, ;
                                                (sel_book) b, ;
                                                (sel_level) c, ;
                                                casdesc d ;
                                        where a.task_no = b.task_no ;
                                                and a.task_no = c.task_no ;
                                        and a.task_no = d.task_no ;
                                        and a.score != 0 ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.oc_cs = moc_cs ;
                                        and a.mission = mmission ;
                                        and a.trng_day = trng_day ;
                                        and a.echelon = sel_echlev ;
                                  order by a.task_no, a.score ;
                                  into CURSOR distrib
                 * 11110
                 case moc_cs != "All" and mmission != "All" and mtrng_day != "All" and
munit_obs != "All" and mechlev = "All"
                            select a.task_no, ;
                                        a.score, ;
                                        d.task_desc ;
                                  from casdata a, ;
                                        (sel_book) b, ;
                                        (sel_level) c, ;
                                        casdesc d ;
                                  where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and a.task_no = d.task_no ;
                                        and a.score != 0 ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.oc_cs = moc_cs ;
                                        and a.mission = mmission ;
                                        and a.trng_day = trng_day ;
                                        and a.unit_obs = munit_obs ;
                                  order by a.task_no, a.score ;
                                  into CURSOR distrib
                 * 11111
                 case moc_cs != "All" and mmission != "All" and mtrng_day != "All" and
munit_obs != "All" and mechlev != "All"
                            select a.task_no, ;
                                        a.score, ;
                                        d.task_desc ;
                                  from casdata a, ;
                                        (sel_book) b, ;
                                        (sel_level) c, ;
                                        casdesc d ;
                                  where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and a.task_no = d.task_no ;
                                        and a.score != 0 ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.oc_cs = moc_cs ;
                                        and a.mission = mmission ;
                                        and a.trng_day = trng_day ;
                                        and a.unit_obs = munit_obs ;
                                        and a.echelon = sel_echlev ;
                                  order by a.task_no, a.score ;
                                  into CURSOR distrib
        endcase
ENDIF

* assign variables default valuse
current_task_no = ""
current_score = 0
score_count = 0

* create another cursor to store the analyzed results
create cursor disrpt ;
        (task_no C(7), ;
                s1 N(2), s2 N(2), s3 N(2), s4 N(2), ;
                s5 N(2), s6 N(2), s7 N(2), s8 N(2), ;
                task_desc C(254))

* make the original cursor the current database
select distrib
* step through the original cursor assigning and analyzing the data
do while .NOT. EOF()
        * get the initialcurrent task number
        current_task_no = distrib.task_no
        * select the new cursor and make a new record
        select disrpt
        append blank
        * store the data
        replace disrpt.task_no with distrib.task_no
        replace disrpt.task_desc with distrib.task_desc
        * go back to the original cursor for more data
        select distrib
        * step through data until the task number changes
        do while current_task_no = task_no
                score_count = 0
                current_score = score
                * count the scores in each area
                do while current_score = score .AND. current_task_no = task_no
```

```
                    score_count = score_count + 1
                    current_score = score
                    skip
          enddo
          * select the new cursor
          select disrpt
          * save the number of scores in the appropriate valued area
          do case
                    * not done
                    case current_score = 1
                              replace s1 with score_count
                    * not adequate
                    case current_score = 2
                              replace s2 with score_count
                    * marginally adequate
                    case current_score = 3
                              replace s3 with score_count
                    * adequate
                    case current_score = 4
                              replace s4 with score_count
                    * superior
                    case current_score = 5
                              replace s5 with score_count
                    * not observed
                    case current_score = 6
                              replace s6 with score_count
                    * not applicable
                    case current_score = 7
                              replace s7 with score_count
                    * not assessed
                    case current_score = 8
                              replace s8 with score_count
          endcase
          * go back to the original for another look
          select distrib
     enddo
enddo

* finalize on the new cursor for the report
select disrpt
```

APPENDIX P

PROGRAM PROVIDING SQL QUERIES FOR THE TASK REMARKS COMPARISON REPORT

1. Program: cascrpt3.prg

2. Author: D.Butterfield, PRC Inc.

3. Date: 20 April 1994

4. Notes: Program produced to generate the necessary SQL query from the selections requested by the user in the casrpts.prg program. Provides the data for the task remarks comparison report. Each case statement is a different SQL query based on the 'All' selections.

5. Usage: Program is called from casrpts.prg. program can not be executed by itself. However, the SQL statements may be used by cut and paste and then replacing the variables with selection criteria.

6. Files: Does not use files, but it does create two temporary CURSOR files for use. The first contains the requested data selections which are then analyzed with results placed into the second CURSOR for use by the report form casrpt3.

7. Problems: None noted.

8. History:

Date	Name	Ver	Modifications	by
04/20/94	cascrpt3	1.0	original	dbb
07/21/94	cascrpt3	1.2	many without detail	dbb
10/06/94	cascrpt3	1.3	added 'All' rotations selection	dbb

```
******************************************************************
* task remarks comparison report
*
* different selection criteria changes report output.
* all of the case statements are near identical except for which
* variable is on and which is not. at each case statement the number
* indicates which of the four variables is an 'All' and which is not.
* indicates which of the four variables is an 'All' and which is not.
* an 'All' is indicated as a '0' and a NOT 'All' is indicated as a '1'.

IF sel_rota != "All"
          do case
                    * 00000
                    case moc_cs = "All" and mmission = "All" and mtrng_day = "All" and
munit_obs = "All" and mechlev = "All"
                              select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                                   from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                                   where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                   order by a.task_no ;
                                   into CURSOR ocremarks
                    * 00001
                    case moc_cs = "All" and mmission = "All" and mtrng_day = "All" and
munit_obs = "All" and mechlev != "All"
                              select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                                   from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                                   where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.echelon = sel_echlev ;
                                   order by a.task_no ;
                                   into CURSOR ocremarks
                    * 00010
                    case moc_cs = "All" and mmission = "All" and mtrng_day = "All" and
munit_obs != "All" and mechlev = "All"
```

```
                    select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                              from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                              where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.unit_obs = munit_obs ;
                              order by a.task_no ;
                              into CURSOR ocremarks
          * 00011
          case moc_cs = "All" and mmission = "All" and mtrng_day = "All" and
munit_obs != "All" and mechlev != "All"
                    select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                              from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                              where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.unit_obs = munit_obs ;
                                        and a.echelon = sel_echlev ;
                              order by a.task_no ;
                              into CURSOR ocremarks
          * 00100
          case moc_cs = "All" and mmission = "All" and mtrng_day != "All" and
munit_obs = "All" and mechlev = "All"
                    select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                              from casdata a, ;
```

```
                                        casrem b, ;
                                        (sel_book) c ;
                              where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.trng_day = mtrng_day ;
                              order by a.task_no ;
                              into CURSOR ocremarks
               * 00101
               case moc_cs = "All" and mmission = "All" and mtrng_day != "All" and
munit_obs = "All" and mechlev != "All"
                         select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                              from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                              where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.trng_day = mtrng_day ;
                                        and a.echelon = sel_echlev ;
                              order by a.task_no ;
                              into CURSOR ocremarks
               * 00110
               case moc_cs = "All" and mmission = "All" and mtrng_day != "All" and
munit_obs != "All" and mechlev = "All"
                         select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                              from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                              where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.trng_day = mtrng_day ;
                                        and a.unit_obs = munit_obs ;
                                  order by a.task_no ;
                                  into CURSOR ocremarks
                  * 00111
                  case moc_cs = "All" and mmission = "All" and mtrng_day != "All" and
munit_obs != "All" and mechlev != "All"
                            select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                                  from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                                  where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.trng_day = mtrng_day ;
                                        and a.unit_obs = munit_obs ;
                                        and a.echelon = sel_echlev ;
                                  order by a.task_no ;
                                  into CURSOR ocremarks
                  * 01000
                  case moc_cs = "All" and mmission != "All" and mtrng_day = "All" and
munit_obs = "All" and mechlev = "All"
                            select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                                  from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                                  where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.mission = mmission ;
                                  order by a.task_no ;
                                  into CURSOR ocremarks
```

```
                    * 01001
                    case moc_cs = "All" and mmission != "All" and mtrng_day = "All" and
munit_obs = "All" and mechlev != "All"
                              select a.task_no, ;
                                                a.mission, ;
                                                a.oc_cs, ;
                                                a.remarks, ;
                                                b.task_desc ;
                                    from casdata a, ;
                                                casrem b, ;
                                                (sel_book) c ;
                                    where a.task_no = b.task_no ;
                                                and a.task_no = c.task_no ;
                                                and b.tk_id_type = sel_type ;
                                                and a.task_id = sel_tkid ;
                                                and a.rotation = sel_rota ;
                                                and a.mission = mmission ;
                                                and a.echelon = sel_echlev ;
                                    order by a.task_no ;
                                    into CURSOR ocremarks
                    * 01010
                    case moc_cs = "All" and mmission != "All" and mtrng_day = "All" and
munit_obs != "All" and mechlev = "All"
                              select a.task_no, ;
                                                a.mission, ;
                                                a.oc_cs, ;
                                                a.remarks, ;
                                                b.task_desc ;
                                    from casdata a, ;
                                                casrem b, ;
                                                (sel_book) c ;
                                    where a.task_no = b.task_no ;
                                                and a.task_no = c.task_no ;
                                                and b.tk_id_type = sel_type ;
                                                and a.task_id = sel_tkid ;
                                                and a.rotation = sel_rota ;
                                                and a.mission = mmission ;
                                                and a.unit_obs = munit_obs ;
                                    order by a.task_no ;
                                    into CURSOR ocremarks
                    * 01011
                    case moc_cs = "All" and mmission != "All" and mtrng_day = "All" and
munit_obs != "All" and mechlev != "All"
                              select a.task_no, ;
                                                a.mission, ;
```

```
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                                from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                                where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.mission = mmission ;
                                        and a.unit_obs = munit_obs ;
                                        and a.echelon = sel_echlev ;
                                order by a.task_no ;
                                into CURSOR ocremarks
                * 01100
                case moc_cs = "All" and mmission != "All" and mtrng_day != "All" and
munit_obs = "All" and mechlev = "All"
                        select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                                from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                                where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.mission = mmission ;
                                        and a.trng_day = mtrng_day;
                                order by a.task_no ;
                                into CURSOR ocremarks
                * 01101
                case moc_cs = "All" and mmission != "All" and mtrng_day != "All" and
munit_obs = "All" and mechlev != "All"
                        select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                                from casdata a, ;
```

```
                                        casrem b, ;
                                        (sel_book) c ;
                                 where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.mission = mmission ;
                                        and a.trng_day = mtrng_day;
                                        and a.echelon = sel_echlev ;
                                 order by a.task_no ;
                                 into CURSOR ocremarks
                  * 01110
                  case moc_cs = "All" and mmission != "All" and mtrng_day != "All" and
munit_obs != "All" and mechlev = "All"
                          select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                                 from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                                 where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.mission = mmission ;
                                        and a.trng_day = mtrng_day;
                                        and a.unit_obs = munit_obs ;
                                 order by a.task_no ;
                                 into CURSOR ocremarks
                  * 01111
                  case moc_cs = "All" and mmission != "All" and mtrng_day != "All" and
munit_obs != "All" and mechlev != "All"
                          select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                                 from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                                 where a.task_no = b.task_no ;
```

```
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.mission = mmission ;
                                        and a.trng_day = mtrng_day;
                                        and a.unit_obs = munit_obs ;
                                        and a.echelon = sel_echlev ;
                                  order by a.task_no ;
                                  into CURSOR ocremarks
                 * 10000
                 case moc_cs != "All" and mmission = "All" and mtrng_day = "All" and
munit_obs = "All" and mechlev = "All"
                           select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                                  from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                                  where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.oc_cs = moc_cs ;
                                  order by a.task_no ;
                                  into CURSOR ocremarks
                 * 10001
                 case moc_cs != "All" and mmission = "All" and mtrng_day = "All" and
munit_obs = "All" and mechlev != "All"
                           select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                                  from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                                  where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.oc_cs = moc_cs ;
                                        and a.echelon = sel_echlev ;
                              order by a.task_no ;
                              into CURSOR ocremarks
              * 10010
              case moc_cs != "All" and mmission = "All" and mtrng_day = "All" and
munit_obs != "All" and mechlev = "All"
                        select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                              from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                              where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.oc_cs = moc_cs ;
                                        and a.unit_obs = munit_obs ;
                              order by a.task_no ;
                              into CURSOR ocremarks
              * 10011
              case moc_cs != "All" and mmission = "All" and mtrng_day = "All" and
munit_obs != "All" and mechlev != "All"
                        select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                              from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                              where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.oc_cs = moc_cs ;
                                        and a.unit_obs = munit_obs ;
                                        and a.echelon = sel_echlev ;
                              order by a.task_no ;
                              into CURSOR ocremarks
```

```
                    * 10100
                    case moc_cs != "All" and mmission = "All" and mtrng_day != "All" and
munit_obs = "All" and mechlev = "All"
                              select a.task_no, ;
                                          a.mission, ;
                                          a.oc_cs, ;
                                          a.remarks, ;
                                          b.task_desc ;
                                    from casdata a, ;
                                          casrem b, ;
                                          (sel_book) c ;
                                    where a.task_no = b.task_no ;
                                          and a.task_no = c.task_no ;
                                          and b.tk_id_type = sel_type ;
                                          and a.task_id = sel_tkid ;
                                          and a.rotation = sel_rota ;
                                          and a.oc_cs = moc_cs ;
                                          and a.trng_day = mtrng_day ;
                                    order by a.task_no ;
                                    into CURSOR ocremarks
                    * 10101
                    case moc_cs != "All" and mmission = "All" and mtrng_day != "All" and
munit_obs = "All" and mechlev != "All"
                              select a.task_no, ;
                                          a.mission, ;
                                          a.oc_cs, ;
                                          a.remarks, ;
                                          b.task_desc ;
                                    from casdata a, ;
                                          casrem b, ;
                                          (sel_book) c ;
                                    where a.task_no = b.task_no ;
                                          and a.task_no = c.task_no ;
                                          and b.tk_id_type = sel_type ;
                                          and a.task_id = sel_tkid ;
                                          and a.rotation = sel_rota ;
                                          and a.oc_cs = moc_cs ;
                                          and a.trng_day = mtrng_day ;
                                          and a.echelon = sel_echlev ;
                                    order by a.task_no ;
                                    into CURSOR ocremarks
                    * 10110
                    case moc_cs != "All" and mmission = "All" and mtrng_day != "All" and
munit_obs != "All" and mechlev = "All"
                              select a.task_no, ;
```

```
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                                  from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                                  where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.oc_cs = moc_cs ;
                                        and a.trng_day = mtrng_day ;
                                        and a.unit_obs = munit_obs ;
                                  order by a.task_no ;
                                  into CURSOR ocremarks
                  * 10111
                  case moc_cs != "All" and mmission = "All" and mtrng_day != "All" and
munit_obs != "All" and mechlev != "All"
                            select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                                  from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                                  where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.oc_cs = moc_cs ;
                                        and a.trng_day = mtrng_day ;
                                        and a.unit_obs = munit_obs ;
                                        and a.echelon = sel_echlev ;
                                  order by a.task_no ;
                                  into CURSOR ocremarks
                  * 11000
                  case moc_cs != "All" and mmission != "All" and mtrng_day = "All" and
munit_obs = "All" and mechlev = "All"
                            select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                              from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                              where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.oc_cs = moc_cs ;
                                        and a.mission = mmission ;
                              order by a.task_no ;
                              into CURSOR ocremarks
            * 11001
            case moc_cs != "All" and mmission != "All" and mtrng_day = "All" and
munit_obs = "All" and mechlev != "All"
                    select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                              from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                              where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.oc_cs = moc_cs ;
                                        and a.mission = mmission ;
                                        and a.echelon = sel_echlev ;
                              order by a.task_no ;
                              into CURSOR ocremarks
            * 11010
            case moc_cs != "All" and mmission != "All" and mtrng_day = "All" and
munit_obs != "All" and mechlev = "All"
                    select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                              from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                              where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.oc_cs = moc_cs ;
                                        and a.mission = mmission ;
                                        and a.unit_obs = munit_obs ;
                              order by a.task_no ;
                              into CURSOR ocremarks
               * 11011
               case moc_cs != "All" and mmission != "All" and mtrng_day = "All" and
munit_obs != "All" and mechlev != "All"
                    select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                              from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                              where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.oc_cs = moc_cs ;
                                        and a.mission = mmission ;
                                        and a.unit_obs = munit_obs ;
                                        and a.echelon = sel_echlev ;
                              order by a.task_no ;
                              into CURSOR ocremarks
               * 11100
               case moc_cs != "All" and mmission != "All" and mtrng_day != "All" and
munit_obs = "All" and mechlev = "All"
                    select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                              from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                              where a.task_no = b.task_no ;
```

```
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.oc_cs = moc_cs ;
                                        and a.mission = mmission ;
                                        and a.trng_day = mtrng_day ;
                                  order by a.task_no ;
                                  into CURSOR ocremarks
                  * 11101
                  case moc_cs != "All" and mmission != "All" and mtrng_day != "All" and
munit_obs = "All" and mechlev != "All"
                            select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                                  from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                                  where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.oc_cs = moc_cs ;
                                        and a.mission = mmission ;
                                        and a.trng_day = mtrng_day ;
                                        and a.echelon = sel_echlev ;
                                  order by a.task_no ;
                                  into CURSOR ocremarks
                  * 11110
                  case moc_cs != "All" and mmission != "All" and mtrng_day != "All" and
munit_obs != "All" and mechlev = "All"
                            select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                                  from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                                  where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.oc_cs = moc_cs ;
                                        and a.mission = mmission ;
                                        and a.trng_day = mtrng_day ;
                                        and a.unit_obs = munit_obs ;
                                order by a.task_no ;
                                into CURSOR ocremarks
                * 11111
                case moc_cs != "All" and mmission != "All" and mtrng_day != "All" and
munit_obs != "All" and mechlev != "All"
                        select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                                from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                                where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.rotation = sel_rota ;
                                        and a.oc_cs = moc_cs ;
                                        and a.mission = mmission ;
                                        and a.trng_day = mtrng_day ;
                                        and a.unit_obs = munit_obs ;
                                        and a.echelon = sel_echlev ;
                                order by a.task_no ;
                                into CURSOR ocremarks
        endcase
ELSE
        do case
                * 00000
                case moc_cs = "All" and mmission = "All" and mtrng_day = "All" and
munit_obs = "All" and mechlev = "All"
                        select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                                from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                                        where a.task_no = b.task_no ;
                                                and a.task_no = c.task_no ;
                                                and b.tk_id_type = sel_type ;
                                                and a.task_id = sel_tkid ;
                                        order by a.task_no ;
                                        into CURSOR ocremarks
                        * 00001
                        case moc_cs = "All" and mmission = "All" and mtrng_day = "All" and
munit_obs = "All" and mechlev != "All"
                                select a.task_no, ;
                                                a.mission, ;
                                                a.oc_cs, ;
                                                a.remarks, ;
                                                b.task_desc ;
                                        from casdata a, ;
                                                casrem b, ;
                                                (sel_book) c ;
                                        where a.task_no = b.task_no ;
                                                and a.task_no = c.task_no ;
                                                and b.tk_id_type = sel_type ;
                                                and a.task_id = sel_tkid ;
                                                and a.echelon = sel_echlev ;
                                        order by a.task_no ;
                                        into CURSOR ocremarks
                        * 00010
                        case moc_cs = "All" and mmission = "All" and mtrng_day = "All" and
munit_obs != "All" and mechlev = "All"
                                select a.task_no, ;
                                                a.mission, ;
                                                a.oc_cs, ;
                                                a.remarks, ;
                                                b.task_desc ;
                                        from casdata a, ;
                                                casrem b, ;
                                                (sel_book) c ;
                                        where a.task_no = b.task_no ;
                                                and a.task_no = c.task_no ;
                                                and b.tk_id_type = sel_type ;
                                                and a.task_id = sel_tkid ;
                                                and a.unit_obs = munit_obs ;
                                        order by a.task_no ;
                                        into CURSOR ocremarks
                        * 00011
                        case moc_cs = "All" and mmission = "All" and mtrng_day = "All" and
munit_obs != "All" and mechlev != "All"
```

```
                    select a.task_no, ;
                                a.mission, ;
                                a.oc_cs, ;
                                a.remarks, ;
                                b.task_desc ;
                          from casdata a, ;
                                casrem b, ;
                                (sel_book) c ;
                          where a.task_no = b.task_no ;
                                and a.task_no = c.task_no ;
                                and b.tk_id_type = sel_type ;
                                and a.task_id = sel_tkid ;
                                and a.unit_obs = munit_obs ;
                                and a.echelon = sel_echlev ;
                          order by a.task_no ;
                          into CURSOR ocremarks
              * 00100
              case moc_cs = "All" and mmission = "All" and mtrng_day != "All" and
munit_obs = "All" and mechlev = "All"
                    select a.task_no, ;
                                a.mission, ;
                                a.oc_cs, ;
                                a.remarks, ;
                                b.task_desc ;
                          from casdata a, ;
                                casrem b, ;
                                (sel_book) c ;
                          where a.task_no = b.task_no ;
                                and a.task_no = c.task_no ;
                                and b.tk_id_type = sel_type ;
                                and a.task_id = sel_tkid ;
                                and a.trng_day = mtrng_day ;
                          order by a.task_no ;
                          into CURSOR ocremarks
              * 00101
              case moc_cs = "All" and mmission = "All" and mtrng_day != "All" and
munit_obs = "All" and mechlev != "All"
                    select a.task_no, ;
                                a.mission, ;
                                a.oc_cs, ;
                                a.remarks, ;
                                b.task_desc ;
                          from casdata a, ;
                                casrem b, ;
                                (sel_book) c ;
```

```
                                        where a.task_no = b.task_no ;
                                                and a.task_no = c.task_no ;
                                                and b.tk_id_type = sel_type ;
                                                and a.task_id = sel_tkid ;
                                                and a.trng_day = mtrng_day ;
                                                and a.echelon = sel_echlev ;
                                        order by a.task_no ;
                                        into CURSOR ocremarks
                        * 00110
                        case moc_cs = "All" and mmission = "All" and mtrng_day != "All" and
munit_obs != "All" and mechlev = "All"
                                select a.task_no, ;
                                                a.mission, ;
                                                a.oc_cs, ;
                                                a.remarks, ;
                                                b.task_desc ;
                                        from casdata a, ;
                                                casrem b, ;
                                                (sel_book) c ;
                                        where a.task_no = b.task_no ;
                                                and a.task_no = c.task_no ;
                                                and b.tk_id_type = sel_type ;
                                                and a.task_id = sel_tkid ;
                                                and a.trng_day = mtrng_day ;
                                                and a.unit_obs = munit_obs ;
                                        order by a.task_no ;
                                        into CURSOR ocremarks
                        * 00111
                        case moc_cs = "All" and mmission = "All" and mtrng_day != "All" and
munit_obs != "All" and mechlev != "All"
                                select a.task_no, ;
                                                a.mission, ;
                                                a.oc_cs, ;
                                                a.remarks, ;
                                                b.task_desc ;
                                        from casdata a, ;
                                                casrem b, ;
                                                (sel_book) c ;
                                        where a.task_no = b.task_no ;
                                                and a.task_no = c.task_no ;
                                                and b.tk_id_type = sel_type ;
                                                and a.task_id = sel_tkid ;
                                                and a.trng_day = mtrng_day ;
                                                and a.unit_obs = munit_obs ;
                                                and a.echelon = sel_echlev ;
                                        order by a.task_no ;
                                        into CURSOR ocremarks
                        * 01000
                        case moc_cs = "All" and mmission != "All" and mtrng_day = "All" and
munit_obs = "All" and mechlev = "All"
                                select a.task_no, ;
                                                a.mission, ;
                                                a.oc_cs, ;
                                                a.remarks, ;
                                                b.task_desc ;
                                        from casdata a, ;
                                                casrem b, ;
                                                (sel_book) c ;
                                        where a.task_no = b.task_no ;
                                                and a.task_no = c.task_no ;
                                                and b.tk_id_type = sel_type ;
                                                and a.task_id = sel_tkid ;
                                                and a.mission = mmission ;
                                        order by a.task_no ;
                                        into CURSOR ocremarks
                        * 01001
                        case moc_cs = "All" and mmission != "All" and mtrng_day = "All" and
munit_obs = "All" and mechlev != "All"
                                select a.task_no, ;
                                                a.mission, ;
                                                a.oc_cs, ;
                                                a.remarks, ;
                                                b.task_desc ;
                                        from casdata a, ;
                                                casrem b, ;
                                                (sel_book) c ;
                                        where a.task_no = b.task_no ;
                                                and a.task_no = c.task_no ;
                                                and b.tk_id_type = sel_type ;
                                                and a.task_id = sel_tkid ;
                                                and a.mission = mmission ;
                                                and a.echelon = sel_echlev ;
                                        order by a.task_no ;
                                        into CURSOR ocremarks
                        * 01010
                        case moc_cs = "All" and mmission != "All" and mtrng_day = "All" and
munit_obs != "All" and mechlev = "All"
                                select a.task_no, ;
                                                a.mission, ;
                                                a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                              from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                              where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.mission = mmission ;
                                        and a.unit_obs = munit_obs ;
                              order by a.task_no ;
                              into CURSOR ocremarks
               * 01011
               case moc_cs = "All" and mmission != "All" and mtrng_day = "All" and
munit_obs != "All" and mechlev != "All"
                         select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                              from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                              where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.mission = mmission ;
                                        and a.unit_obs = munit_obs ;
                                        and a.echelon = sel_echlev ;
                              order by a.task_no ;
                              into CURSOR ocremarks
               * 01100
               case moc_cs = "All" and mmission != "All" and mtrng_day != "All" and
munit_obs = "All" and mechlev = "All"
                         select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                              from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                              where a.task_no = b.task_no ;
```

```
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.mission = mmission ;
                                        and a.trng_day = mtrng_day;
                                  order by a.task_no ;
                                  into CURSOR ocremarks
                  * 01101
                  case moc_cs = "All" and mmission != "All" and mtrng_day != "All" and
munit_obs = "All" and mechlev != "All"
                           select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                                  from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                                  where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.mission = mmission ;
                                        and a.trng_day = mtrng_day;
                                        and a.echelon = sel_echlev ;
                                  order by a.task_no ;
                                  into CURSOR ocremarks
                  * 01110
                  case moc_cs = "All" and mmission != "All" and mtrng_day != "All" and
munit_obs != "All" and mechlev = "All"
                           select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                                  from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                                  where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.mission = mmission ;
                                        and a.trng_day = mtrng_day;
                                        and a.unit_obs = munit_obs ;
                                        order by a.task_no ;
                                        into CURSOR ocremarks
                        * 01111
                        case moc_cs = "All" and mmission != "All" and mtrng_day != "All" and
munit_obs != "All" and mechlev != "All"
                                select a.task_no, ;
                                                a.mission, ;
                                                a.oc_cs, ;
                                                a.remarks, ;
                                                b.task_desc ;
                                        from casdata a, ;
                                                casrem b, ;
                                                (sel_book) c ;
                                        where a.task_no = b.task_no ;
                                                and a.task_no = c.task_no ;
                                                and b.tk_id_type = sel_type ;
                                                and a.task_id = sel_tkid ;
                                                and a.mission = mmission ;
                                                and a.trng_day = mtrng_day;
                                                and a.unit_obs = munit_obs ;
                                                and a.echelon = sel_echlev ;
                                        order by a.task_no ;
                                        into CURSOR ocremarks
                        * 10000
                        case moc_cs != "All" and mmission = "All" and mtrng_day = "All" and
munit_obs = "All" and mechlev = "All"
                                select a.task_no, ;
                                                a.mission, ;
                                                a.oc_cs, ;
                                                a.remarks, ;
                                                b.task_desc ;
                                        from casdata a, ;
                                                casrem b, ;
                                                (sel_book) c ;
                                        where a.task_no = b.task_no ;
                                                and a.task_no = c.task_no ;
                                                and b.tk_id_type = sel_type ;
                                                and a.task_id = sel_tkid ;
                                                and a.oc_cs = moc_cs ;
                                        order by a.task_no ;
                                        into CURSOR ocremarks
                        * 10001
                        case moc_cs != "All" and mmission = "All" and mtrng_day = "All" and
munit_obs = "All" and mechlev != "All"
                                select a.task_no, ;
```

```
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                              from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                              where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.oc_cs = moc_cs ;
                                        and a.echelon = sel_echlev ;
                              order by a.task_no ;
                              into CURSOR ocremarks
          * 10010
          case moc_cs != "All" and mmission = "All" and mtrng_day = "All" and
munit_obs != "All" and mechlev = "All"
                    select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                              from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                              where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.oc_cs = moc_cs ;
                                        and a.unit_obs = munit_obs ;
                              order by a.task_no ;
                              into CURSOR ocremarks
          * 10011
          case moc_cs != "All" and mmission = "All" and mtrng_day = "All" and
munit_obs != "All" and mechlev != "All"
                    select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                              from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
```

```
                                   where a.task_no = b.task_no ;
                                         and a.task_no = c.task_no ;
                                         and b.tk_id_type = sel_type ;
                                         and a.task_id = sel_tkid ;
                                         and a.oc_cs = moc_cs ;
                                         and a.unit_obs = munit_obs ;
                                         and a.echelon = sel_echlev ;
                                   order by a.task_no ;
                                   into CURSOR ocremarks
                     * 10100
                     case moc_cs != "All" and mmission = "All" and mtrng_day != "All" and
munit_obs = "All" and mechlev = "All"
                              select a.task_no, ;
                                         a.mission, ;
                                         a.oc_cs, ;
                                         a.remarks, ;
                                         b.task_desc ;
                                   from casdata a, ;
                                         casrem b, ;
                                         (sel_book) c ;
                                   where a.task_no = b.task_no ;
                                         and a.task_no = c.task_no ;
                                         and b.tk_id_type = sel_type ;
                                         and a.task_id = sel_tkid ;
                                         and a.oc_cs = moc_cs ;
                                         and a.trng_day = mtrng_day ;
                                   order by a.task_no ;
                                   into CURSOR ocremarks
                     * 10101
                     case moc_cs != "All" and mmission = "All" and mtrng_day != "All" and
munit_obs = "All" and mechlev != "All"
                              select a.task_no, ;
                                         a.mission, ;
                                         a.oc_cs, ;
                                         a.remarks, ;
                                         b.task_desc ;
                                   from casdata a, ;
                                         casrem b, ;
                                         (sel_book) c ;
                                   where a.task_no = b.task_no ;
                                         and a.task_no = c.task_no ;
                                         and b.tk_id_type = sel_type ;
                                         and a.task_id = sel_tkid ;
                                         and a.oc_cs = moc_cs ;
                                         and a.trng_day = mtrng_day ;
```

```
                                        and a.echelon = sel_echlev ;
                              order by a.task_no ;
                              into CURSOR ocremarks
            * 10110
            case moc_cs != "All" and mmission = "All" and mtrng_day != "All" and
munit_obs != "All" and mechlev = "All"
                        select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                              from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                              where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.oc_cs = moc_cs ;
                                        and a.trng_day = mtrng_day ;
                                        and a.unit_obs = munit_obs ;
                              order by a.task_no ;
                              into CURSOR ocremarks
            * 10111
            case moc_cs != "All" and mmission = "All" and mtrng_day != "All" and
munit_obs != "All" and mechlev != "All"
                        select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                              from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                              where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.oc_cs = moc_cs ;
                                        and a.trng_day = mtrng_day ;
                                        and a.unit_obs = munit_obs ;
                                        and a.echelon = sel_echlev ;
                              order by a.task_no ;
                              into CURSOR ocremarks
            * 11000
```

```
                    case moc_cs != "All" and mmission != "All" and mtrng_day = "All" and
munit_obs = "All" and mechlev = "All"
                              select a.task_no, ;
                                                  a.mission, ;
                                                  a.oc_cs, ;
                                                  a.remarks, ;
                                                  b.task_desc ;
                                        from casdata a, ;
                                                  casrem b, ;
                                                  (sel_book) c ;
                                        where a.task_no = b.task_no ;
                                                  and a.task_no = c.task_no ;
                                                  and b.tk_id_type = sel_type ;
                                                  and a.task_id = sel_tkid ;
                                                  and a.oc_cs = moc_cs ;
                                                  and a.mission = mmission ;
                                        order by a.task_no ;
                                        into CURSOR ocremarks
                    * 11001
                    case moc_cs != "All" and mmission != "All" and mtrng_day = "All" and
munit_obs = "All" and mechlev != "All"
                              select a.task_no, ;
                                                  a.mission, ;
                                                  a.oc_cs, ;
                                                  a.remarks, ;
                                                  b.task_desc ;
                                        from casdata a, ;
                                                  casrem b, ;
                                                  (sel_book) c ;
                                        where a.task_no = b.task_no ;
                                                  and a.task_no = c.task_no ;
                                                  and b.tk_id_type = sel_type ;
                                                  and a.task_id = sel_tkid ;
                                                  and a.oc_cs = moc_cs ;
                                                  and a.mission = mmission ;
                                                  and a.echelon = sel_echlev ;
                                        order by a.task_no ;
                                        into CURSOR ocremarks
                    * 11010
                    case moc_cs != "All" and mmission != "All" and mtrng_day = "All" and
munit_obs != "All" and mechlev = "All"
                              select a.task_no, ;
                                                  a.mission, ;
                                                  a.oc_cs, ;
                                                  a.remarks, ;
                                        b.task_desc ;
                              from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                              where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.oc_cs = moc_cs ;
                                        and a.mission = mmission ;
                                        and a.unit_obs = munit_obs ;
                              order by a.task_no ;
                              into CURSOR ocremarks
               * 11011
               case moc_cs != "All" and mmission != "All" and mtrng_day = "All" and
munit_obs != "All" and mechlev != "All"
                         select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                              from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                              where a.task_no = b.task_no ;
                                        and a.task_no = c.task_no ;
                                        and b.tk_id_type = sel_type ;
                                        and a.task_id = sel_tkid ;
                                        and a.oc_cs = moc_cs ;
                                        and a.mission = mmission ;
                                        and a.unit_obs = munit_obs ;
                                        and a.echelon = sel_echlev ;
                              order by a.task_no ;
                              into CURSOR ocremarks
               * 11100
               case moc_cs != "All" and mmission != "All" and mtrng_day != "All" and
munit_obs = "All" and mechlev = "All"
                         select a.task_no, ;
                                        a.mission, ;
                                        a.oc_cs, ;
                                        a.remarks, ;
                                        b.task_desc ;
                              from casdata a, ;
                                        casrem b, ;
                                        (sel_book) c ;
                                        where a.task_no = b.task_no ;
                                                and a.task_no = c.task_no ;
                                                and b.tk_id_type = sel_type ;
                                                and a.task_id = sel_tkid ;
                                                and a.oc_cs = moc_cs ;
                                                and a.mission = mmission ;
                                                and a.trng_day = mtrng_day ;
                                        order by a.task_no ;
                                        into CURSOR ocremarks
                        * 11101
                        case moc_cs != "All" and mmission != "All" and mtrng_day != "All" and
munit_obs = "All" and mechlev != "All"
                                select a.task_no, ;
                                                a.mission, ;
                                                a.oc_cs, ;
                                                a.remarks, ;
                                                b.task_desc ;
                                        from casdata a, ;
                                                casrem b, ;
                                                (sel_book) c ;
                                        where a.task_no = b.task_no ;
                                                and a.task_no = c.task_no ;
                                                and b.tk_id_type = sel_type ;
                                                and a.task_id = sel_tkid ;
                                                and a.oc_cs = moc_cs ;
                                                and a.mission = mmission ;
                                                and a.trng_day = mtrng_day ;
                                                and a.echelon = sel_echlev ;
                                        order by a.task_no ;
                                        into CURSOR ocremarks
                        * 11110
                        case moc_cs != "All" and mmission != "All" and mtrng_day != "All" and
munit_obs != "All" and mechlev = "All"
                                select a.task_no, ;
                                                a.mission, ;
                                                a.oc_cs, ;
                                                a.remarks, ;
                                                b.task_desc ;
                                        from casdata a, ;
                                                casrem b, ;
                                                (sel_book) c ;
                                        where a.task_no = b.task_no ;
                                                and a.task_no = c.task_no ;
                                                and b.tk_id_type = sel_type ;
                                                and a.task_id = sel_tkid ;
```

```
                                                  and a.oc_cs = moc_cs ;
                                                  and a.mission = mmission ;
                                                  and a.trng_day = mtrng_day ;
                                                  and a.unit_obs = munit_obs ;
                                        order by a.task_no ;
                                        into CURSOR ocremarks
                        * 11111
                        case moc_cs != "All" and mmission != "All" and mtrng_day != "All" and
munit_obs != "All" and mechlev != "All"
                                select a.task_no, ;
                                                  a.mission, ;
                                                  a.oc_cs, ;
                                                  a.remarks, ;
                                                  b.task_desc ;
                                        from casdata a, ;
                                                  casrem b, ;
                                                  (sel_book) c ;
                                        where a.task_no = b.task_no ;
                                                  and a.task_no = c.task_no ;
                                                  and b.tk_id_type = sel_type ;
                                                  and a.task_id = sel_tkid ;
                                                  and a.oc_cs = moc_cs ;
                                                  and a.mission = mmission ;
                                                  and a.trng_day = mtrng_day ;
                                                  and a.unit_obs = munit_obs ;
                                                  and a.echelon = sel_echlev ;
                                        order by a.task_no ;
                                        into CURSOR ocremarks
        endcase
ENDIF

* create a new cursor to put only the 'good' remarks in
create cursor remrpt ;
        (task_no C(7), mission C(15), oc_cs C(5), ;
                remarks M, task_desc C(254))

* get the original cursor data
select ocremarks
* analyze until end of cursor data
do while .NOT. EOF()
        * check the remarks field for valid data
        if NOT EMPTY(remarks)
                * valid data found so save it in new cursor
                select remrpt
                append blank
                replace task_no with ocremarks.task_no
                replace mission with ocremarks.mission
                replace oc_cs with ocremarks.oc_cs
                replace remarks with ocremarks.remarks
                replace task_desc with ocremarks.task_desc
        endif
        * return to original cursor data
        select ocremarks
        * get next record
        skip
enddo

* default to new cursor for report
select remrpt
```

APPENDIX Q

PROGRAM TO INFORM THE USER THAT NO DATA WAS FOUND PER HIS QUERY REQUEST

1. Program: casnone.prg

2. Author: D.Butterfield, PRC Inc.

3. Date: 21 April 1994

4. Notes: Program produced to provide the user with a popup message which indicates no data was found using the users selection criteria.

5. Usage: Program is called from the casrpts.prg program after one of the SQL cascrpt?.prg programs finds no data.

6. Files: Does not use other files.

7. Problems: None noted.

8. History:

Date	Name	Ver	Modifications	By
04/21/94	casnone	1.0	original	dbb
07/21/94	casnone	1.1	no detail	dbb

```
**********************************************************************

#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
        SET TALK OFF
        m.talkstat = "ON"
ELSE
        m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

m.rborder = SET("READBORDER")
```

```
SET READBORDER ON

m.currarea = SELECT()


*       ****************************************************
*       *
*       *           Windows Window definitions
*       *
*       ****************************************************
*


IF NOT WEXIST("_nonefound")
        DEFINE WINDOW _nonefound ;
                AT  0.000, 0.000  ;
                SIZE 11.154,80.600 ;
                FONT "MS Sans Serif", 8 ;
                FLOAT ;
                CLOSE ;
                NOMINIMIZE
        MOVE WINDOW _nonefound CENTER
ENDIF


*       ****************************************************
*       *
*       *           CASNONE/Windows Screen Layout
*       *
*       ****************************************************
*


#REGION 1
IF WVISIBLE("_nonefound")
        ACTIVATE WINDOW _nonefound SAME
ELSE
        ACTIVATE WINDOW _nonefound NOSHOW
ENDIF

@ 6.846,31.200 GET mokay ;
        PICTURE "@*HT OK" ;
        SIZE 1.769,14.167,0.667 ;
        DEFAULT 1 ;
        FONT "MS Sans Serif", 8 ;
        STYLE "B"
@ 4.000,7.800 SAY "No Information Found Using the Selected Items"  ;
```

```
          FONT "MS Sans Serif", 10 ;
          STYLE "BT"
@ 3.154,5.200 TO 6.154,75.400 ;
          PEN 2, 8
@ 1.846,1.600 TO 9.384,79.000 ;
          PEN 2, 8

IF NOT WVISIBLE("_nonefound")
          ACTIVATE WINDOW _nonefound
ENDIF

READ

RELEASE WINDOW _nonefound
SELECT (m.currarea)


#REGION 0

SET READBORDER &rborder

IF m.talkstat = "ON"
          SET TALK ON
ENDIF
IF m.compstat = "ON"
          SET COMPATIBLE ON
ENDIF
```

APPENDIX R

PROGRAM TO ALLOW USER INTERFACE TO SQL QUERY AND REPORTS FOR OUTCOME REPORTS

1. Program: casorpts.prg
2. Author: D.Butterfield/Jerry Fargo, PRC Inc.
3. Date: 15 August 1994
4. Notes:
5. Usage:
6. Files:
7. Problems:
8. History:

Date	Name	Ver	Modifications	By
08/15/94	casorpts	1.0	original	dbb

9. Last modified: 11/18/94 at 12:25:12
10. Procedures and Functions _MAKE_ARRAYS
    _DOREPORTS()
11. Set by: _QU70J2FEC (procedure in CASMENU.MPR)
12. Calls:

_MAKE_ARRAYS	(procedure in CASORPTS.PRG)
_DOREPORTS()	(function in CASORPTS.PRG)
_WIN_LOWER()	(function in CASENTRY.PRG)

```
*****************************************************************

CLOSE DATABASES

DIMENSION r_array[1,1]
DIMENSION m_array[1,1]
DIMENSION t_array[1,1]

DO _make_arrays

#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
   SET TALK OFF
   m.talkstat = "ON"
ELSE
   m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

m.rborder = SET("READBORDER")
SET readborder ON

m.currarea = SELECT()


*       ********************************************************
*       *               Windows Window definitions
*       ********************************************************
IF NOT WEXIST("_outcome")
   DEFINE WINDOW _outcome ;
      AT  0.000, 0.000  ;
      SIZE 15.769,67.600 ;
      FONT "MS Sans Serif", 8 ;
      FLOAT ;
      CLOSE ;
      MINIMIZE ;
      SYSTEM
   MOVE WINDOW _outcome CENTER
ENDIF


*       ********************************************************
*       *
*       *             CASORPTS/Windows Screen Layout
*       *
*       ********************************************************
*
```

```
#REGION 1
IF WVISIBLE("_outcome")
  ACTIVATE WINDOW _outcome SAME
ELSE
  ACTIVATE WINDOW _outcome NOSHOW
ENDIF
* overall window title and underline
@ 0.615,7.200 SAY "CAS Outcome Database Reports" ;
  FONT "MS Sans Serif", 12 ;
  STYLE "BT"
@ 2.154,7.200 TO 2.154,60.200 ;
  PEN 1, 8 ;
  STYLE "1"
* titles/headings
@ 2.846,2.400 SAY "Report Selection" ;
  FONT "MS Sans Serif", 8 ;
  STYLE "BT"
@ 2.846,43.200 SAY "Rotation" ;
  FONT "MS Sans Serif", 8 ;
  STYLE "BT"
@ 5.846,43.200 SAY "Mission" ;
  FONT "MS Sans Serif", 8 ;
  STYLE "BT"
@ 8.846,43.200 SAY "Training Day" ;
  FONT "MS Sans Serif", 8 ;
  STYLE "BT"
* report selection
@ 4.769,4.800 GET mreport ;
  PICTURE "@*RVN Rotation Summary;Mission Summary;Day Summary;Comments
Summary" ;
  SIZE 1.308,23.000,0.308 ;
  DEFAULT 1 ;
  FONT "MS Sans Serif", 8 ;
  STYLE "BT"
* database content selections
* rotation
@ 4.000,43.200 GET mrot_sel ;
  PICTURE "@^" ;
  FROM r_array ;
  SIZE 1.538,18.167 ;
  DEFAULT "All" ;
  FONT "MS Sans Serif", 8 ;
  STYLE "B"
* mission
@ 7.000,43.200 GET mmis_sel ;
```

```
  PICTURE "@^" ;
  FROM m_array ;
  SIZE 1.538,18.167 ;
  DEFAULT "All" ;
  FONT "MS Sans Serif", 8 ;
  STYLE "B"
* training day
@ 10.000,43.200 GET mtda_sel ;
  PICTURE "@^" ;
  FROM t_array ;
  SIZE 1.538,18.167 ;
  DEFAULT "All" ;
  FONT "MS Sans Serif", 8 ;
  STYLE "B"
* preview/print buttons
@ 12.923,4.800 GET mpreview ;
  PICTURE "@*HN Preview" ;
  SIZE 1.769,9.667,0.667 ;
  DEFAULT 0 ;
  FONT "MS Sans Serif", 8 ;
  STYLE "B" ;
  VALID _doreports()
@ 12.923,28.000 GET mprint ;
  PICTURE "@*HN Print" ;
  SIZE 1.769,9.667,0.667 ;
  DEFAULT 0 ;
  FONT "MS Sans Serif", 8 ;
  STYLE "B" ;
  VALID _doreports()
* exit window button
@ 12.923,51.200 GET mcancel ;
  PICTURE "@*HT Cancel" ;
  SIZE 1.769,9.667,0.667 ;
  DEFAULT 0 ;
  FONT "MS Sans Serif", 8 ;
  STYLE "B"

IF NOT WVISIBLE("_outcome")
  ACTIVATE WINDOW _outcome
ENDIF

READ DEACTIVATE _win_lower()

RELEASE WINDOW _outcome
SELECT (m.currarea)
```

```
#REGION 0

SET readborder &rborder

IF m.talkstat = "ON"
   SET TALK ON
ENDIF
IF m.compstat = "ON"
   SET COMPATIBLE ON
ENDIF

*******************************************************************
*!*****************************************************************************
*
*!
*!      Procedure: _MAKE_ARRAYS
*!
*!      Called by: CASORPTS.PRG
*!
*!           Uses: CASOUTC.DBF
*!
*!*****************************************************************************
*
PROCEDURE _make_arrays
SELECT DISTINCT rotation FROM casoutc INTO ARRAY temp_array
* insert an 'All' selection into the array list
m.count = _TALLY
DIMENSION temp_array(m.count + 1, 1)
= AINS(temp_array, m.count + 1)
temp_array[m.count+1] = "All"

= ACOPY(temp_array, r_array)

SELECT DISTINCT mission FROM casoutc INTO ARRAY temp_array
* insert an 'All' selection into the array list
m.count = _TALLY
DIMENSION temp_array(m.count + 1, 1)
= AINS(temp_array, m.count + 1)
temp_array[m.count+1] = "All"

= ACOPY(temp_array, m_array)

SELECT DISTINCT trn_day FROM casoutc INTO ARRAY temp_array
* insert an 'All' selection into the array list
```

```
m.count = _TALLY
DIMENSION temp_array(m.count + 1, 1)
= AINS(temp_array, m.count + 1)
temp_array[m.count+1] = "All"

= ACOPY(temp_array, t_array)
RETURN

*!*************************************************************************
*
*!
*!       Function: _DOREPORTS
*!
*!      Called by: CASORPTS.PRG
*!
*!          Calls: CASOROT.PRG
*!
*!*************************************************************************
*
FUNCTION _doreports
valid_data = .F.
DO CASE
CASE mreport = 1
   * Rotation Summary
   DO casorot WITH mrot_sel, mmis_sel, mtda_sel
CASE mreport = 2
   * Mission Summary
   DO casorot WITH mrot_sel, mmis_sel, mtda_sel
CASE mreport = 3
   * Day Summary
   DO casorot WITH mrot_sel, mmis_sel, mtda_sel
CASE mreport = 4
   * Comments Summary
   DO casorot WITH mrot_sel, mmis_sel, mtda_sel
ENDCASE
mpreview = 0
mprint = 0
RETURN

*!*************************************************************************
*
*!
*!       Function: _WIN_LOWER
*!
*!      Called by: CASENTRY.PRG
*!              : CASOUT1.PRG
*!              : CASORPTS.PRG
*!
*!*************************************************************************
*
FUNCTION _win_lower
RETURN .F.


*: EOF: CASORPTS.PRG
```

APPENDIX S

PROGRAM PROVIDING SQL QUERIES FOR THE OUTCOME REPORTS

1. Program: asorot.prg

2. Author: D.Butterfield/Jerry Fargo, PRC Inc.

3. Date: 15 August 1994

4. Notes:

5. Usage:

6. Files:

7. Problems:

8. History:

Date	Name	Ver	Modifications	By
08/15/94	casorot	1.0	original	dbb

9. Last modified: 11/22/94 at 9:51:52

10. Procedures and Functions:
    _ROT_SUMMARY
    _MIS_SUMMARY
    _DAY_SUMMARY
    _CMT_SUMMARY

11. Set by: _DOREPORTS() (function in CASORPTS.PRG)

12. Calls:

_ROT_SUMMARY	(procedure in CASOROT.PRG)
_MIS_SUMMARY	(procedure in CASOROT.PRG)
_DAY_SUMMARY	(procedure in CASOROT.PRG)
_CMT_SUMMARY	(procedure in CASOROT.PRG)
CASNONE.PRG	

13. Uses: CASOUTC.DBF

```
*********************************************************************

PARAMETER sel_rotation, sel_mission, sel_tday

CLOSE DATABASES

DO CASE
  * 000
CASE sel_tday = 'All' AND sel_mission = 'All' AND sel_rotation = 'All'
  SELECT a.rotation, a.mission, a.trn_day, a.oc_cs, ;
    a.leth_a, a.leth_b, a.surv_a, a.surv_b, ;
    a.com_mis, a.com_ene, a.com_tro, a.com_ter, a.com_tim, ;
    a.rem_mis, a.rem_ene, a.rem_tro, a.rem_ter, a.rem_tim ;
    FROM casoutc A ;
    INTO CURSOR castemp
  * 001
CASE sel_tday = 'All' AND sel_mission = 'All' AND sel_rotation != 'All'
  SELECT a.rotation, a.mission, a.trn_day, a.oc_cs,  ;
    a.leth_a, a.leth_b, a.surv_a, a.surv_b, ;
    a.com_mis, a.com_ene, a.com_tro, a.com_ter, a.com_tim, ;
    a.rem_mis, a.rem_ene, a.rem_tro, a.rem_ter, a.rem_tim ;
    FROM casoutc A ;
    WHERE a.rotation = sel_rotation ;
    INTO CURSOR castemp
  * 010
CASE sel_tday = 'All' AND sel_mission != 'All' AND sel_rotation = 'All'
  SELECT a.rotation, a.mission, a.trn_day, a.oc_cs, ;
    a.leth_a, a.leth_b, a.surv_a, a.surv_b, ;
    a.com_mis, a.com_ene, a.com_tro, a.com_ter, a.com_tim, ;
    a.rem_mis, a.rem_ene, a.rem_tro, a.rem_ter, a.rem_tim ;
    FROM casoutc A ;
    WHERE a.mission = sel_mission ;
    INTO CURSOR castemp
  * 011
CASE sel_tday = 'All' AND sel_mission != 'All' AND sel_rotation != 'All'
  SELECT a.rotation, a.mission, a.trn_day, a.oc_cs, ;
    a.leth_a, a.leth_b, a.surv_a, a.surv_b, ;
    a.com_mis, a.com_ene, a.com_tro, a.com_ter, a.com_tim, ;
    a.rem_mis, a.rem_ene, a.rem_tro, a.rem_ter, a.rem_tim ;
    FROM casoutc A ;
    WHERE a.mission = sel_mission ;
    AND a.rotation = sel_rotation ;
    INTO CURSOR castemp
  * 100
CASE sel_tday != 'All' AND sel_mission = 'All' AND sel_rotation = 'All'
```

```
  SELECT a.rotation, a.mission, a.trn_day, a.oc_cs, ;
    a.leth_a, a.leth_b, a.surv_a, a.surv_b, ;
    a.com_mis, a.com_ene, a.com_tro, a.com_ter, a.com_tim, ;
    a.rem_mis, a.rem_ene, a.rem_tro, a.rem_ter, a.rem_tim ;
    FROM casoutc A ;
    WHERE a.trn_day = sel_tday ;
    INTO CURSOR castemp
  * 101
CASE sel_tday != 'All' AND sel_mission = 'All' AND sel_rotation != 'All'
  SELECT a.rotation, a.mission, a.trn_day, a.oc_cs, ;
    a.leth_a, a.leth_b, a.surv_a, a.surv_b, ;
    a.com_mis, a.com_ene, a.com_tro, a.com_ter, a.com_tim, ;
    a.rem_mis, a.rem_ene, a.rem_tro, a.rem_ter, a.rem_tim ;
    FROM casoutc A ;
    WHERE a.trn_day = sel_tday ;
    AND a.rotation = sel_rotation ;
    INTO CURSOR castemp
  * 110
CASE sel_tday != 'All' AND sel_mission != 'All' AND sel_rotation = 'All'
  SELECT a.rotation, a.mission, a.trn_day, a.oc_cs, ;
    a.leth_a, a.leth_b, a.surv_a, a.surv_b, ;
    a.com_mis, a.com_ene, a.com_tro, a.com_ter, a.com_tim, ;
    a.rem_mis, a.rem_ene, a.rem_tro, a.rem_ter, a.rem_tim ;
    FROM casoutc A ;
    WHERE a.trn_day = sel_tday ;
    AND a.mission = sel_mission ;
    INTO CURSOR castemp
  * 111
CASE sel_tday != 'All' AND sel_mission != 'All' AND sel_rotation != 'All'
  SELECT a.rotation, a.mission, a.trn_day, a.oc_cs, ;
    a.leth_a, a.leth_b, a.surv_a, a.surv_b, ;
    a.com_mis, a.com_ene, a.com_tro, a.com_ter, a.com_tim, ;
    a.rem_mis, a.rem_ene, a.rem_tro, a.rem_ter, a.rem_tim ;
    FROM casoutc A ;
    WHERE a.trn_day = sel_tday ;
    AND a.mission = sel_mission ;
    AND a.rotation = sel_rotation ;
    INTO CURSOR castemp
ENDCASE

IF RECCOUNT() > 0
  DO CASE
    * CAS Rotation Summary
  CASE mreport = 1
    DO _rot_summary
    * CAS Mission Summary
  CASE mreport = 2
    DO _mis_summary
    * CAS Day Summary
  CASE mreport = 3
    DO _day_summary
    * CAS Comment Summary
  CASE mreport = 4
    DO _cmt_summary
  ENDCASE
ELSE
  DO casnone
ENDIF

RETURN

*!*********************************************************************
*
*!
*!      Procedure: _ROT_SUMMARY
*!
*!      Called by: CASOROT.PRG
*!
*!          Calls: CASLOC.PRG
*!
*!   Report Forms: CASOROT.FRX
*!
*!*********************************************************************
*
PROCEDURE _rot_summary
num_mis = RECCOUNT()
mrotation = sel_rotation
mtrn_day = sel_tday

mleth_a = 0
mleth_b = 0
msurv_a = 0
msurv_b = 0

mycom_mis = 0
mycom_ene = 0
mycom_tro = 0
mycom_ter = 0
mycom_tim = 0
```

```
mncom_mis = 0
mncom_ene = 0
mncom_tro = 0
mncom_ter = 0
mncom_tim = 0

GO TOP
DO WHILE !EOF()
  * total number of weapons used
  mleth_a = leth_a + mleth_a

  * total number of vehicles used
  mleth_b = leth_b + mleth_b

  * total number of aircraft starting mission
  msurv_a = surv_a + msurv_a

  * total number of aircraft at the end of all missions
  msurv_b = surv_b + msurv_b

  * mission
  IF com_mis = 1
    mycom_mis = mycom_mis + 1
  ELSE
    mncom_mis = mncom_mis + 1
  ENDIF

  * enemy
  IF com_ene = 1
    mycom_ene = mycom_ene + 1
  ELSE
    mncom_ene = mncom_ene + 1
  ENDIF

  * troops
  IF com_tro = 1
    mycom_tro = mycom_tro + 1
  ELSE
    mncom_tro = mncom_tro + 1
  ENDIF

  * terrain
  IF com_ter = 1
    mycom_ter = mycom_ter + 1
  ELSE
```

```
     mncom_ter = mncom_ter + 1
   ENDIF

   * time
   IF com_tim = 1
     mycom_tim = mycom_tim + 1
   ELSE
     mncom_tim = mncom_tim + 1
   ENDIF

   SKIP
ENDDO

* create a cursor to store the analyzed results and
* to produce one page of report vs one page per record
CREATE CURSOR rptout ;
   (   rota C(4), ;
   tmis N(4), ;
   tday C(8), ;
   tleth_a N(4), tleth_b N(4), ;
   tsurv_a N(4), tsurv_b N(4), ;
   tycom_mis N(2), tycom_ene N(2), tycom_tro N(2), ;
   tycom_ter N(2), tycom_tim N(2), ;
   tncom_mis N(2), tncom_ene N(2), tncom_tro N(2), ;
   tncom_ter N(2), tncom_tim N(2) ;
   )

APPEND BLANK
REPLACE rota WITH mrotation
REPLACE tmis WITH num_mis
REPLACE tday WITH mtrn_day

REPLACE tleth_a WITH mleth_a
REPLACE tleth_b WITH mleth_b
REPLACE tsurv_a WITH msurv_a
REPLACE tsurv_b WITH msurv_b

REPLACE tycom_mis WITH mycom_mis
REPLACE tycom_ene WITH mycom_ene
REPLACE tycom_tro WITH mycom_tro
REPLACE tycom_ter WITH mycom_ter
REPLACE tycom_tim WITH mycom_tim

REPLACE tncom_mis WITH mncom_mis
REPLACE tncom_ene WITH mncom_ene
REPLACE tncom_tro WITH mncom_tro
REPLACE tncom_ter WITH mncom_ter
REPLACE tncom_tim WITH mncom_tim

* DO CASE
* CASE mpreview = 1
*   REPORT FORM casorot PREVIEW
* CASE mprint = 1
*    SET CONSOLE OFF
*    REPORT FORM casorot TO PRINTER
*    SET CONSOLE ON
* ENDCASE

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*          REPORT PRINT/FILE
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
mlocation = 1
mokay =1
mrfile = "ROTSUM.TXT"
IF mpreview = 1
   REPORT FORM casorot PREVIEW
ELSE
   DO casloc WITH mlocation, mokay, mrfile
   IF mokay = 1
      IF mlocation = 1
         SET CONSOLE OFF
         REPORT FORM casorot TO PRINTER
         SET CONSOLE ON
      ELSE
         SET CONSOLE OFF
         REPORT FORM casorot TO FILE (mrfile)
         SET CONSOLE ON
      ENDIF
   ENDIF
   RETURN

*!**************************************************************************
*
*!
*!     Procedure: _MIS_SUMMARY
*!
*!     Called by: CASOROT.PRG
*!
*!         Calls: CASLOC.PRG
*!
```

```
*!  Report Forms: CASOMIS.FRX
*!
*!*************************************************************************
*
PROCEDURE _mis_summary
mmission = sel_mission
mrotation = sel_rotation
mtrn_day = sel_tday

mleth_a = 0
mleth_b = 0
msurv_a = 0
msurv_b = 0

mycom_mis = 0
mycom_ene = 0
mycom_tro = 0
mycom_ter = 0
mycom_tim = 0

mncom_mis = 0
mncom_ene = 0
mncom_tro = 0
mncom_ter = 0
mncom_tim = 0

mrem_mis = rem_mis
mrem_ene = rem_ene
mrem_tro = rem_tro
mrem_ter = rem_ter
mrem_tim = rem_tim

GO TOP
DO WHILE !EOF()
   * total number of weapons used
   mleth_a = leth_a + mleth_a

   * total number of vehicles used
   mleth_b = leth_b + mleth_b

   * total number of aircraft starting mission
   msurv_a = surv_a + msurv_a

   * total number of aircraft at the end of all missions
   msurv_b = surv_b + msurv_b
```

```
    * mission
    IF com_mis = 1
      mycom_mis = mycom_mis + 1
    ELSE
      mncom_mis = mncom_mis + 1
    ENDIF
    * enemy
    IF com_ene = 1
      mycom_ene = mycom_ene + 1
    ELSE
      mncom_ene = mncom_ene + 1
    ENDIF
    * troops
    IF com_tro = 1
      mycom_tro = mycom_tro + 1
    ELSE
      mncom_tro = mncom_tro + 1
    ENDIF

    * terrain
    IF com_ter = 1
      mycom_ter = mycom_ter + 1
    ELSE
      mncom_ter = mncom_ter + 1
    ENDIF

    * time
    IF com_tim = 1
      mycom_tim = mycom_tim + 1
    ELSE
      mncom_tim = mncom_tim + 1
    ENDIF

    SKIP
ENDDO

* create another cursor to store the analyzed results
CREATE CURSOR rptout ;
    (   rota C(4), ;
    tmis C(10), ;
    tday C(4), ;
    tleth_a N(4), tleth_b N(4), ;
    tsurv_a N(4), tsurv_b N(4), ;
    tycom_mis N(2), tycom_ene N(2), tycom_tro N(2), ;
```

```
   tycom_ter N(2), tycom_tim N(2), ;
   tncom_mis N(2), tncom_ene N(2), tncom_tro N(2), ;
   tncom_ter N(2), tncom_tim N(2), ;
   trem_mis m, trem_ene m, trem_tro m, ;
   trem_ter m, trem_tim m;
   )

APPEND BLANK
REPLACE rota WITH mrotation
REPLACE tmis WITH mmission
REPLACE tday WITH mtrn_day

REPLACE tleth_a WITH mleth_a
REPLACE tleth_b WITH mleth_b
REPLACE tsurv_a WITH msurv_a
REPLACE tsurv_b WITH msurv_b

REPLACE tycom_mis WITH mycom_mis
REPLACE tycom_ene WITH mycom_ene
REPLACE tycom_tro WITH mycom_tro
REPLACE tycom_ter WITH mycom_ter
REPLACE tycom_tim WITH mycom_tim

REPLACE tncom_mis WITH mncom_mis
REPLACE tncom_ene WITH mncom_ene
REPLACE tncom_tro WITH mncom_tro
REPLACE tncom_ter WITH mncom_ter
REPLACE tncom_tim WITH mncom_tim

REPLACE trem_mis WITH mrem_mis
REPLACE trem_ene WITH mrem_ene
REPLACE trem_tro WITH mrem_tro
REPLACE trem_ter WITH mrem_ter
REPLACE trem_tim WITH mrem_tim

* IF mpreview = 1
*   REPORT FORM casomis PREVIEW
* ELSE
*   SET CONSOLE OFF
*   REPORT FORM casomis TO PRINTER
*   SET CONSOLE ON
* ENDIF
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*         REPORT PRINT/FILE
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

```
mlocation = 1
mokay =1
mrfile = "MISSUM.TXT"
IF mpreview = 1
   REPORT FORM casomis PREVIEW
ELSE
   DO casloc WITH mlocation, mokay, mrfile
   IF mokay = 1
      IF mlocation = 1
         SET CONSOLE OFF
         REPORT FORM casomis TO PRINTER
         SET CONSOLE ON
      ELSE
         SET CONSOLE OFF
         REPORT FORM casomis TO FILE (mrfile)
         SET CONSOLE ON
      ENDIF
   ENDIF
   RETURN

*!*****************************************************************************
*
*!
*!      Procedure: _DAY_SUMMARY
*!
*!      Called by: CASOROT.PRG
*!
*!          Calls: CASLOC.PRG
*!
*!   Report Forms: CASODAY.FRX
*!
*!*****************************************************************************
*
PROCEDURE _day_summary
num_miss = RECCOUNT()
mrotation = sel_rotation
mtrn_day = sel_tday

mleth_a = 0
mleth_b = 0
msurv_a = 0
msurv_b = 0

mycom_mis = 0
mycom_ene = 0
```

```
mycom_tro = 0
mycom_ter = 0
mycom_tim = 0

mncom_mis = 0
mncom_ene = 0
mncom_tro = 0
mncom_ter = 0
mncom_tim = 0

GO TOP
DO WHILE !EOF()
  * total number of weapons used
  mleth_a = leth_a + mleth_a

  * total number of vehicles used
  mleth_b = leth_b + mleth_b

  * total number of aircraft starting mission
  msurv_a = surv_a + msurv_a

  * total number of aircraft at the end of all missions
  msurv_b = surv_b + msurv_b

  * mission
  IF com_mis = 1
    mycom_mis = mycom_mis + 1
  ELSE
    mncom_mis = mncom_mis + 1
  ENDIF

  * enemy
  IF com_ene = 1
    mycom_ene = mycom_ene + 1
  ELSE
    mncom_ene = mncom_ene + 1
  ENDIF

  * troops
  IF com_tro = 1
    mycom_tro = mycom_tro + 1
  ELSE
    mncom_tro = mncom_tro + 1
  ENDIF
  * terrain
  IF com_ter = 1
     mycom_ter = mycom_ter + 1
  ELSE
     mncom_ter = mncom_ter + 1
  ENDIF

  * time
  IF com_tim = 1
     mycom_tim = mycom_tim + 1
  ELSE
     mncom_tim = mncom_tim + 1
  ENDIF

  SKIP
ENDDO

* create a cursor to store the analyzed results
CREATE CURSOR rptout ;
  (   rota C(4), ;
  tmis N(4), ;
  tday C(4), ;
  tleth_a N(4), tleth_b N(4), ;
  tsurv_a N(4), tsurv_b N(4), ;
  tycom_mis N(2), tycom_ene N(2), tycom_tro N(2), ;
  tycom_ter N(2), tycom_tim N(2), ;
  tncom_mis N(2), tncom_ene N(2), tncom_tro N(2), ;
  tncom_ter N(2), tncom_tim N(2) ;
  )

APPEND BLANK
REPLACE rota WITH mrotation
REPLACE tmis WITH num_miss
REPLACE tday WITH mtrn_day

REPLACE tleth_a WITH mleth_a
REPLACE tleth_b WITH mleth_b
REPLACE tsurv_a WITH msurv_a
REPLACE tsurv_b WITH msurv_b

REPLACE tycom_mis WITH mycom_mis
REPLACE tycom_ene WITH mycom_ene
REPLACE tycom_tro WITH mycom_tro
REPLACE tycom_ter WITH mycom_ter
REPLACE tycom_tim WITH mycom_tim
REPLACE tncom_mis WITH mncom_mis
REPLACE tncom_ene WITH mncom_ene
REPLACE tncom_tro WITH mncom_tro
REPLACE tncom_ter WITH mncom_ter
REPLACE tncom_tim WITH mncom_tim

* IF mpreview = 1
*    REPORT FORM casoday PREVIEW
* ELSE
*    REPORT FORM casoday TO PRINTER
* ENDIF
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*          REPORT PRINT/FILE
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
mlocation = 1
mokay =1
mrfile = "DAYSUM.TXT"
IF mpreview = 1
  REPORT FORM casoday PREVIEW
ELSE
  DO casloc WITH mlocation, mokay, mrfile
  IF mokay = 1
    IF mlocation = 1
      SET CONSOLE OFF
      REPORT FORM casoday TO PRINTER
      SET CONSOLE ON
    ELSE
      SET CONSOLE OFF
      REPORT FORM casoday TO FILE (mrfile)
      SET CONSOLE ON
    ENDIF
  ENDIF
  RETURN

*!***********************************************************************
*
*!
*!     Procedure: _CMT_SUMMARY
*!
*!     Called by: CASOROT.PRG
*!
*!         Calls: CASLOC.PRG
*!
*!   Report Forms: CASOCMT.FRX
*!
```

```
*!*************************************************************************
*
PROCEDURE _cmt_summary
* IF mpreview = 1
*    REPORT FORM casocmt PREVIEW
* ELSE
*    SET CONSOLE OFF
*    REPORT FORM casocmt TO PRINTER
*    SET CONSOLE ON
*ENDIF
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*          REPORT PRINT/FILE
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
mlocation = 1
mokay =1
mrfile = "CMTSUM.TXT"
IF mpreview = 1
  REPORT FORM casocmt PREVIEW
ELSE
  DO casloc WITH mlocation, mokay, mrfile
  IF mokay = 1
    IF mlocation = 1
      SET CONSOLE OFF
      REPORT FORM casocmt TO PRINTER
      SET CONSOLE ON
    ELSE
      SET CONSOLE OFF
      REPORT FORM casocmt TO FILE (mrfile)
      SET CONSOLE ON
    ENDIF
  ENDIF
  RETURN

*: EOF: CASOROT.PRG
```

APPENDIX T

PROGRAM TO CONVERT ECI DATA TO CAS DATABASE FORMAT

1. Program: casconv.prg

2. Author: D.Butterfield, PRC Inc.

3. Date: 12 April 1994

4. Notes: Program developed to provide a user interface to the ECI import/conversion process. User executes the program by selecting ECI convert from the Close Air Support Menu item displayed when casproj.app is executed. Program reads in, converts and stores ECI CAS files to the CASDATA.DBF or CASOUTC.DBF database file.

5. Usage: Select ECI convert from casproj.app menu. Program can be executed stand alone if the type of conversion, which relates directly to the database name, casdata or casoutc is passed as a parameter.

6. Files: Uses the selected ECI files to convert into the CASDATA or CASOUTC databases.

7. Problems: User/operator needs to set the CTC letter for the location the ECI data is being received from. to set the letter modify the variable mctc_letter below by changing the mctc_letter to one of the following:

CTC	variable	letter
---	--------	------
NTC	mctc_letter =	N
CMTC	mctc_letter =	C
JRTC	mctc_letter =	J

8. History:

Date	Name	Ver	Modifications	By
04/12/94	casconv	1.0	original	dbb
07/21/94	casconv	1.5	many without detail	dbb

```
**********************************************************************
PARAMETERS dconvert

* CHANGE CTC LETTER FOR mctc_letter AT THIS LOCATION
* select initial CTC letter for rotation
* where N = NTC, C = CMTC, J = JRTC
mctc_letter = 'N'

* close all databases and clear screen
close databases
clear

ON ERROR DO caserr WITH ERROR(), MESSAGE()

* open files non-exclusively
SET EXCLUSIVE OFF

* reprocessing of unsuccessful locks is automatic
SET REPROCESS TO AUTOMATIC

#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
        SET TALK OFF
        m.talkstat = "ON"
ELSE
        m.talkstat = "OFF"
ENDIF

m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

m.rborder = SET("READBORDER")
SET READBORDER ON

m.currarea = SELECT()

* create a temporary database cursor to read/append data into
* afacprep.txt (6) JRTC J945
* manintpl.txt (3) JRTC J945
* manintpr.txt (3) JRTC J945
* manfspl.txt  (2) JRTC J945
* manfspr.txt  (2) JRTC J945
* the new ECI default text file should be only 2 columns
*           1st column is the field identifier
*           2nd column is the data
* so only two columns will be used

* set the import database to null
* to force a selection box for the user
import_dbase = ""

SELECT 0

* have the user select the ECI database to convert
import_dbase = GETFILE('DBF', 'Select the ECI database.')

if EMPTY(import_dbase)
   RETURN
else
   USE (import_dbase) ;
                        ALIAS import_dbase;
                        ORDER 0
endif

SET ORDER TO 0

* move the ECI database data to an array
scatter memo to eci_array

* set the number of fields
num_fields = fcount()

* create a new array to manipulate/modify the data
dimension data_array(num_fields, 2)

* fill the new array with the ECI data
for i = 1 to num_fields
            data_array(i, 1) = field(i)
            data_array(i, 2) = eci_array(1, i)
next i
* sort the array data
= asort(data_array)

* close the import ECI database
USE

SELECT 0
```

```
* Convert the ECI databases
DO CASE dconvert
***************************************************************
* append to the casdata database
***************************************************************
case dconvert = "casdata"

*
* initialize the data database for use
*
current_dbase = "casdata.dbf"

IF USED(current_dbase)
          SELECT current_dbase
          SET ORDER TO 0
ELSE
          SELECT 0

          USE (LOCFILE(current_dbase,"DBF","Where is CASDATA.DBF?")) ;
                              AGAIN ALIAS current_dbase ;
                              ORDER 0
ENDIF

do _waitwindow with current_dbase, import_dbase

SET ORDER TO 0

* obtain number of rows and columns in array
num_rows = alen(data_array, 1)
num_cols = alen(data_array, 2)

* set initial row, column, and column count
row_ptr = 1
col_ptr = 1
col_count = 1

* create new data records from disk data
do while col_count < num_cols
          * initialize and 'zero out' memory variables
          store '' to mtime
          store '' to mrotation
          store '' to mcas_mis
          store '' to mmission
          store '' to munit_obs
          store '' to mechelon
          store '' to moc_cs
          store '' to mtrng_day
          store '' to mtask_id
          store '' to mtask_no
          store  0 to mscore
          store '' to mremarks

          mod_tk_no = .F.
          mod_REM = .F.
          init_fields = .T.

          * since we can not forsee the structure/order of the ECIinput
          * we must step through the text file twice to obtain the data.
          * first, obtain the constant fields for the database.
          do while row_ptr <= num_rows
                    current_field = upper(data_array(row_ptr,col_ptr))

                    do case
                              case left(current_field, 3) = 'ECI'
                                        * skip ECI fields

                              case current_field = 'MISSION'
                                        * the ECI samples had nothing in this field
                                        store data_array(row_ptr, col_ptr + col_count) to
mcas_mis

                              case current_field = 'ECHELON'
                                        store upper(data_array(row_ptr, col_ptr + col_count))
to mechelon
                              case current_field = 'FIELD115'
                                        store upper(data_array(row_ptr, col_ptr + col_count))
to mechelon
                              case current_field = 'OCID'
                                        store upper(data_array(row_ptr, col_ptr + col_count))
to moc_cs

                              case current_field = 'UNITMISS'
                                        store upper(data_array(row_ptr, col_ptr + col_count))
to mmission

                              case current_field = 'ROTATION'
                                        ch_psn = atc('-', data_array(row_ptr, col_ptr +
col_count))

                                        if ch_psn != 0
                                                  current_field =
stuff(data_array(row_ptr,col_ptr + col_count),ch_psn,1,'')
                                                    do case
                                                            case substr(current_field, 3, 1) =
'0'
                                                                    current_field =
stuff(current_field, 3, 1,'')
                                                            case substr(current_field, 3, 2) =
'10'
                                                                    current_field =
stuff(current_field, 3, 1,'A')
                                                            case substr(current_field, 3, 2) =
'11'
                                                                    current_field =
stuff(current_field, 3, 1,'B')
                                                            case substr(current_field, 3, 2) =
'12'
                                                                    current_field =
stuff(current_field, 3, 1,'C')
                                                            case substr(current_field, 3, 2) =
'13'
                                                                    current_field =
stuff(current_field, 3, 1,'D')
                                                            case substr(current_field, 3, 2) =
'14'
                                                                    current_field =
stuff(current_field, 3, 1,'E')
                                                            case substr(current_field, 3, 2) =
'15'
                                                                    current_field =
stuff(current_field, 3, 1,'F')
                                                    endcase
                                                    mrotation = stuff(current_field, 1, 0,
mctc_letter)
                                            else
                                                    mrotation = trim(data_array(row_ptr,
col_ptr + col_count))
                                            endif

                                    case current_field = 'UNIT'
                                            store trim(data_array(row_ptr, col_ptr + col_count))
to munit_obs

                                    case current_field = 'TRAINDAY'
                                            * so convert the ECI numeric trainday to character
                                            store transform(data_array(row_ptr, col_ptr +
col_count),'99') to mtrng_day
                    endcase

                    * increment to next field row
                    row_ptr = row_ptr + 1
          * end first pass (row_ptr <= num_rows)
          enddo

          * reset the row pointer for the second pass
          row_ptr = 1

          * step through the text data a second time.
          * obtain the task numbers, scores, and remarks
          do while row_ptr <= num_rows
                    current_field = upper(data_array(row_ptr,col_ptr))

                    * we have all the constant fields so skip those
                    do case
                              case left(current_field, 3) = 'ECI'
                                        * skip ECI fields
                              case current_field = 'MISSION'
                                        * skip this field
                              case left(current_field,5) = 'FIELD'
                                        * skip this field
                              case current_field = 'ECHELON'
                                        * skip this field
                              case current_field = 'OCID'
                                        * skip this field
                              case current_field = 'UNITMISS'
                                        * skip this field
                              case current_field = 'ROTATION'
                                        * skip this field
                              case current_field = 'UNIT'
                                        * skip this field
                              case current_field = 'TRAINDAY'
                                        * skip this field
                              otherwise
                                        * assign proper task identification for database entry
                                        do _get_taskid with current_field, mtask_id,
mod_tk_no, mod_rem

                                        * get the task number field, check for modification
                                        if mod_tk_no
                                                  current_field =
upper(stuff(trim(data_array(row_ptr, col_ptr)), 2, 1, ''))
```

```
                                        else
                                                current_field =
upper(trim(data_array(row_ptr, col_ptr)))
                                        endif

                                        * check for score or remarks data
                                        do _chk_taskno with ;
                                                        current_field, data_array,
row_ptr, col_ptr, ;
                                                        col_count, mremarks, mtask_no,
mscore, mod_rem

                                        * put the data into the database
                                        do _put_fields with ;
                                                        mrotation, mtrng_day, mtime,
munit_obs, ;
                                                        mechelon, moc_cs, mmission,
mcas_mis, ;
                                                        mtask_id, mtask_no, mscore,
mremarks

                                        * null out the temp remarks field
                                        store '' to mremarks
                        endcase

                        * increment to next field row
                        row_ptr = row_ptr + 1
                * end the second pass (row_ptr <= num_rows)
                enddo

                * increment col_count
                col_count = col_count + 1

                * reset row to make another pass
                row_ptr = 1

* end of all columns of data
* end of conversion
enddo

**************************************************************
* append to the casoutc database
**************************************************************
case dconvert = "casoutc"
        *
```

```
* initialize the data database for use
*
current_dbase = "casoutc.dbf"

IF USED(current_dbase)
        SELECT current_dbase
        SET ORDER TO 0
ELSE
        SELECT 0

        USE (LOCFILE(current_dbase,"DBF","Where is CASOUTC.DBF?")) ;
                AGAIN ALIAS current_dbase ;
                ORDER 0
ENDIF

do _waitwindow with current_dbase, import_dbase

SET ORDER TO 0

* obtain number of rows and columns in array
num_rows = alen(data_array, 1)
num_cols = alen(data_array, 2)

* set initial row, column, and column count
row_ptr = 1
col_ptr = 1
col_count = 1

* initialize and 'zero out' memory variables
store '' to mrotation
store '' to mmission
store '' to moc_cs
store '' to mdtg
store 0 to mleth_a
store 0 to mleth_b
store 0 to msurv_a
store 0 to msurv_b
store 0 to mcom_mis
store 0 to mcom_ene
store 0 to mcom_tro
store 0 to mcom_ter
store 0 to mcom_tim
store '' to mrem_mis
store '' to mrem_end
store '' to mrem_tro
```

```
    store " to mrem_ter
    store " to mrem_tim

    * step through the ECI data and modify it if necessary
    do while row_ptr <= num_rows
        current_field = upper(data_array(row_ptr,col_ptr))

        do case
            case left(current_field, 3) = 'ECI'
                * skip ECI fields
            case left(current_field,5) = 'FIELD'
                * skip this field
            case current_field = 'ROTATION'
                ch_psn = atc('-', data_array(row_ptr, col_ptr +
col_count))
                if ch_psn != 0
                    current_field =
stuff(data_array(row_ptr,col_ptr + col_count),ch_psn,1,")
                    do case
                        case substr(current_field, 3, 1) =
'0'
                            current_field =
stuff(current_field, 3, 1,")
                        case substr(current_field, 3, 2) =
'10'
                            current_field =
stuff(current_field, 3, 1,'A')
                        case substr(current_field, 3, 2) =
'11'
                            current_field =
stuff(current_field, 3, 1,'B')
                        case substr(current_field, 3, 2) =
'12'
                            current_field =
stuff(current_field, 3, 1,'C')
                        case substr(current_field, 3, 2) =
'13'
                            current_field =
stuff(current_field, 3, 1,'D')
                        case substr(current_field, 3, 2) =
'14'
                            current_field =
stuff(current_field, 3, 1,'E')
                        case substr(current_field, 3, 2) =
'15'
```

```
                                                            current_field =
stuff(current_field, 3, 1,'F')
                                              endcase
                                              mrotation = stuff(current_field, 1, 0,
mctc_letter)
                                    else
                                              mrotation = trim(data_array(row_ptr,
col_ptr + col_count))
                                    endif
                          case current_field = 'OC_CS'
                                    store upper(data_array(row_ptr, col_ptr + col_count))
to moc_cs
                          case current_field = 'DTG'
                                    store upper(data_array(row_ptr, col_ptr + col_count))
to mdtg
                          case current_field = 'MISSION'
                                    store upper(data_array(row_ptr, col_ptr + col_count))
to mmission
                          case current_field = 'LETH_A'
                                    store data_array(row_ptr, col_ptr + col_count) to
mleth_a
                          case current_field = 'LETH_B'
                                    store data_array(row_ptr, col_ptr + col_count) to
mleth_b
                          case current_field = 'SURV_A'
                                    store data_array(row_ptr, col_ptr + col_count) to
msurv_a
                          case current_field = 'SURV_B'
                                    store data_array(row_ptr, col_ptr + col_count) to
msurv_b
                          case current_field = 'COM_MIS'
                                    store data_array(row_ptr, col_ptr + col_count) to
mcom_mis
                          case current_field = 'COM_ENE'
                                    store data_array(row_ptr, col_ptr + col_count) to
mcom_ene
                          case current_field = 'COM_TRO'
                                    store data_array(row_ptr, col_ptr + col_count) to
mcom_tro
                          case current_field = 'COM_TER'
                                    store data_array(row_ptr, col_ptr + col_count) to
mcom_ter
                          case current_field = 'COM_TIM'
                                    store data_array(row_ptr, col_ptr + col_count) to
mcom_tim
```

```
                                        case current_field = 'REM_MIS'
                                                  store upper(data_array(row_ptr, col_ptr + col_count))
to mrem_mis
                                        case current_field = 'REM_ENE'
                                                  store upper(data_array(row_ptr, col_ptr + col_count))
to mrem_ene
                                        case current_field = 'REM_TRO'
                                                  store upper(data_array(row_ptr, col_ptr + col_count))
to mrem_tro
                                        case current_field = 'REM_TER'
                                                  store upper(data_array(row_ptr, col_ptr + col_count))
to mrem_ter
                                        case current_field = 'REM_TIM'
                                                  store upper(data_array(row_ptr, col_ptr + col_count))
to mrem_tim
                              endcase
                              * increment to next field row
                              row_ptr = row_ptr + 1
          * end first pass (row_ptr <= num_rows)
          enddo

          * have all of the data assigned/converted
          * store it to the OUTCOME database
          append blank
          replace rotation with mrotation
          replace mission with mmission
          replace oc_cs with moc_cs
          replace dtg with mdtg
          replace leth_a with mleth_a
          replace leth_b with mleth_b
          replace surv_a with msurv_a
          replace surv_b with msurv_b
          replace com_mis with mcom_mis
          replace com_ene with mcom_ene
          replace com_tro with mcom_tro
          replace com_ter with mcom_ter
          replace com_tim with mcom_tim
          replace rem_mis with mrem_mis
          replace rem_ene with mrem_ene
          replace rem_tro with mrem_tro
          replace rem_ter with mrem_ter
          replace rem_tim with mrem_tim

          * end of conversion
endcase
```

```
IF WEXIST('_waitabit')
        RELEASE WINDOW _waitabit
ENDIF

*
*               Windows Closing Databases
*
IF USED(current_dbase)
        SELECT current_dbase
        USE
ENDIF

SELECT (m.currarea)

#REGION 0

SET READBORDER &rborder

IF m.talkstat = "ON"
        SET TALK ON
ENDIF
IF m.compstat = "ON"
        SET COMPATIBLE ON
ENDIF

*
* close all databases in use so next input screen will
* not find difficulty in opening like databases
close databases

* reset on error routine to default
ON ERROR

*************************************************************
* start procedures
*************************************************************
PROCEDURE _get_taskid
PARAMETERS current_field, mtask_id, mod_tk_no, mod_rem
        do case
                case left(current_field, 1) = 'A'
                        mtask_id = 'A'
                case left(current_field, 1) = 'G'
                        if  left(current_field, 2) = 'GA'
                                mtask_id = 'GA'
                                mod_rem = .T.
```

```
            else
                mtask_id = 'G'
            endif
        case left(current_field, 1) = 'M'
            mod_tk_no = .T.
            do case
                case left(current_field, 2) = 'MO'
                    mtask_id = 'MO'
                case left(current_field, 2) = 'MF'
                    mtask_id = 'MF'
                case left(current_field, 2) = 'ML'
                    mtask_id = 'ML'
                case left(current_field, 2) = 'MD'
                    mtask_id = 'MD'
                case left(current_field, 2) = 'MV'
                    mtask_id = 'MV'
                case left(current_field, 2) = 'MG'
                    mtask_id = 'MG'
            endcase
    endcase
RETURN

PROCEDURE _chk_taskno
PARAMETERS current_field, data_array, row_ptr, col_ptr, ;
            col_count, mremarks, mtask_no, mscore, mod_rem

        * check for the remarks (text) field
        if right(current_field, 3) = 'CMT'
            if mod_rem
                mtask_no = stuff(current_field, 5, 3, 'REM')
            else
                mtask_no = stuff(current_field, 4, 3, 'REM')
            endif
            store 0 to mscore
            store trim(data_array(row_ptr, col_ptr + col_count)) to
mremarks
        else
            * not a remark so it is a task_no and score
            store current_field to mtask_no
            * make sure non-remark task number scores are not 0
            if data_array(row_ptr, col_ptr + col_count) = 0
                * ECI stores a 0 for NOT ASSESSED, make it an 8
                mscore = 8
            else
                mscore = data_array(row_ptr, col_ptr + col_count)
```

```
                        endif
                endif
RETURN

PROCEDURE _put_fields
PARAMETERS mrotation, mtrng_day, mtime, munit_obs, ;
                        mechelon, moc_cs, mmission, mcas_mis, ;
                        mtask_id, mtask_no, mscore, mremarks

                * initial fields found and first data obtained
                append blank
                * store initial field values
                replace rotation with mrotation
                replace trng_day with mtrng_day
                replace time with mtime
                replace unit_obs with munit_obs
                replace echelon with mechelon
                replace oc_cs with moc_cs
                replace mission with mmission
                replace cas_mis with mcas_mis
                * store the task, score, and remarks data
                replace task_id with mtask_id
                replace task_no with mtask_no
                replace score with mscore
                replace remarks with mremarks
RETURN

PROCEDURE _waitwindow
PARAMETERS current_dbase, import_dbase

        IF NOT WEXIST("_waitabit")
                DEFINE WINDOW _waitabit ;
                        AT  0.000, 0.000  ;
                SIZE 23.125,120.600 ;
                        TITLE "Close Air Support" ;
                        FONT "Times New Roman", 10 ;
                        FLOAT ;
                        CLOSE ;
                        MINIMIZE ;
                        NONE
                MOVE WINDOW _waitabit CENTER
        ENDIF

        IF WVISIBLE("_waitabit")
                ACTIVATE WINDOW _waitabit SAME
```

```
          ELSE
                    ACTIVATE WINDOW _waitabit NOSHOW
          ENDIF

          @ 5.250,19.400 GET nothing ;
                    PICTURE "@*HT Processing ECI Data" ;
                    SIZE 5.216,24.059,0.235 ;
                    DEFAULT 1 ;
                    FONT "MS Sans Serif", 24 ;
                    STYLE "B"
          @ 4.438,15.600 TO 18.126,105.000 ;
                    PEN 2, 8
                    @ 19.500,20.000 SAY "Wait a moment while the ECI database
<"+import_dbase+"> is converted and" ;
                    FONT "Times New Roman", 10 ;
                    STYLE "B"
          @ 20.500,28.000 SAY "stored into the Close Air Support <"+current_dbase+">
database." ;
                    FONT "Times New Roman", 10 ;
                    STYLE "B"

          IF NOT WVISIBLE("_waitabit")
                    ACTIVATE WINDOW _waitabit
          ENDIF
RETURN
```

APPENDIX U

PROGRAM TO ALLOW SELECTION TO PRINT REPORT TO PRINTER OR TO FILE

1. Program: casloc.prg

2. Author: Dave Butterfield/Jerry Fargo

3. Date: 07/25/94

4. Documented: 10:24:54

5. Set by:

_CREATE_RPT()	(function in CASRPTS.PRG)
_ROT_SUMMARY	(procedure in CASOROT.PRG)
_MIS_SUMMARY	(procedure in CASOROT.PRG)
_DAY_SUMMARY	(procedure in CASOROT.PRG)
_CMT_SUMMARY	(procedure in CASOROT.PRG)

6. Description: This program was automatically generated by GENSCRN.

```
*********************************************************

PARAMETERS mlocation, mokay, mrfile

#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
  SET TALK OFF
  m.talkstat = "ON"
ELSE
  m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

m.rborder = SET("READBORDER")
SET readborder ON
```

```
m.currarea = SELECT()

*       ******************************************************
*       *
*       *                Windows Window definitions
*       *
*       ******************************************************
*

IF NOT WEXIST("_rptlocation")
  DEFINE WINDOW _rptlocation ;
    AT  0.000, 0.000  ;
    SIZE 11.154,46.200 ;
    FONT "MS Sans Serif", 8 ;
    FLOAT ;
    NOCLOSE ;
    MINIMIZE ;
    SYSTEM
  MOVE WINDOW _rptlocation CENTER
ENDIF


*       ******************************************************
*       *
*       *               CASLOC/Windows Screen Layout
*       *
*       ******************************************************
*

#REGION 1
IF WVISIBLE("_rptlocation")
  ACTIVATE WINDOW _rptlocation SAME
ELSE
  ACTIVATE WINDOW _rptlocation NOSHOW
ENDIF
@ 3.692,4.800 GET mlocation ;
  PICTURE "@*RVN Print to the Default Printer;Save to file "+mrfile ;
  SIZE 1.308,32.167,0.308 ;
  DEFAULT 1 ;
  FONT "MS Sans Serif", 8 ;
  STYLE "BT"
@ 7.308,9.600 GET mokay ;
  PICTURE "@*HT OK;Cancel" ;
  SIZE 1.769,10.167,0.667 ;
  DEFAULT 1 ;
  FONT "MS Sans Serif", 8 ;
  STYLE "B"
@ 0.923,2.400 TO 10.231,43.400 ;
  PEN 1, 8
@ 1.923,7.600 SAY "Select Location for Output:" ;
  FONT "MS Sans Serif", 8 ;
  STYLE "BT"

IF NOT WVISIBLE("_rptlocation")
  ACTIVATE WINDOW _rptlocation
ENDIF

READ CYCLE

RELEASE WINDOW _rptlocation
SELECT (m.currarea)


#REGION 0

SET readborder &rborder

IF m.talkstat = "ON"
  SET TALK ON
ENDIF
IF m.compstat = "ON"
  SET COMPATIBLE ON
ENDIF
*: EOF: CASLOC.PRG
```

APPENDIX V

AIR GROUND TRAINING AND FEEDBACK SYSTEM DATABASE

1. Attached are two discs which contain both the collective database developed as a result of this project and the test database used to evaluate the data collection system.

2. The disc marked "casfinal" is the final database with all related programs, ready to be installed and used, once data collection starts. This database is not the test database that contains the data collected during the two validation rotations. There are no data in the two primary CAS databases: CAS Master Database (casdata.dbf) and CAS Outcome Database (casoutc.dbf).

3. The disc marked "castest" is the test database that was used during the two validation rotations. It contains the data that was colllected and used to verify to utility of the different data collection and assessment methods. This data is located in the CAS Master Database (casdata.dbf) and CAS Outcome Database (casoutc.dbf) in "castest.app".

4. Because of the size of the files written for the databases, the files were compressed in order to fit on the discs. Software to decompress the files is included on the discs and is automatically started by the installation procedure. The users must have FoxPro for Windows version 2.5 or 2.6 software loaded on the computer or the network on which these databases are to be loaded.

5. Instructions to load the databases are included as a text file on each disc and are reproduced below.

    - CASFINAL/CASINSTL.TXT contains instructions for installation of the (empty) CAS Final Database, and readying it for acceptance of new data.

    - CASTEST/CASINSTL.TXT contains instructions for installation of the CAS Test Database, which has the test data loaded.

6. CASFINAL\CASINSTL.TXT: Close Air Support Installation Directions

Installation Information

This notepad window, which is displaying the CAS Final installation directions, will probably need to be reduced in size or moved to view the installation window. Once you can see the installation window, select it by placing the mouse cursor anywhere in the window and clicking on it with mouse button one. Then select the "OK" button by tabbing to it and pressing "Enter", or by pressing on it using mouse button one.

This installation will place all of the Close Air Support application programs and databases in the directory C:/CASFINAL.

After the installation is finished, a Windows Group call "Windows Applications" will be created. You will have to install an icon in order to execute the CASFINAL application.

Installing the Icon

1. Click mouse button one in the "Windows Applications" group box to select it.

2. From the Program Manager select the "File" option at the top left hand corner of the Program Manger window by using mouse button one, or the keyboard combination "Alt-F". The drop down File Selection Menu will appear.

3. Select "New..." from the File Selection Menu using mouse button one or the "Alt-N" keyboard combination.

4. A popup window that has the title "New Program Object" will appear. Select "Program Item" by clicking on the title "Program Item" or on the radio button at the left of the title "Program Item". You may also use the keyboard combination "Alt-I" to select "Program Item".

5. Using mouse button one or by tabbing (use the "tab" key), push or press "Enter" to select the "OK" button on the popup menu. The "New Program Object" popup window will disappear.

6. A new popup window will appear on the screen with the window title, "Program Item Properties". Please follow the these steps to fill in the

required information:

a. Select the "Description:" area either by using mouse button one or by using the "Alt-D" combination from the keyboard. Type in "CASFINAL" or "CAS Final" in the text area.

b. Select the "Command Line:" area by using mouse button one, the "Alt-C" combination, or the "Tab" key. Type in the following according to your FoxPro for Windows version:

   For FoxPro for Windows Version 2.6, type in:

   ```
   C:\FPW26\FOXPROW.EXE CASFINAL.APP
   ```

   For FoxPro for Windows Version 2.5, type in:

   ```
   C:\FOXPROW\FOXPROW.EXE CASFINAL.APP
   ```

Note: *If you are using a network version of FoxPro for Windows, replace the above drive and path for FoxPro for Windows with your drive and path. For example, if you are using FoxPro for Windows Version 2.5 and it is located on the F: drive under the APPS directory, you would type in the following for your "Command Line:" input:*

```
F:\APPS\FPW26\FOXPROW.EXE CASFINAL.APP
```

c. Next select the "Working Directory:" area by using mouse button one, keyboard "Alt-W" combination, or the "Tab" key. Type in the following directory path:

   ```
   C:\CASFINAL
   ```

d. You are finished with the typing. Now click on the "OK" button in the popup window. A FoxPro for Windows icon (a fox head) should appear in the "Windows Applications" group with the title CASFINAL or CAS Final below it.

7. The Icon installation is complete. Double click on the icon and within a few moments you should have FoxPro for Windows initialize and execute the CASFINAL application.

7. CASTEST\CASINSTL.TXT: Close Air Support Installation Directions

Installation Information

The installation instructions for installing the CASTEST database are the same as the instructions for the CASFINAL database, except that all references to "CASFINAL" are replaced with "CASTEST".